UNIVERSITÀ DEGLI STUDI FIRENZE

DOTTORATO DI RICERCA IN INGEGNERIA INDUSTRIALE

CICLO XXVII

Determining product architecture during the conceptual design phase

Settore Scientifico Disciplinare ING/IND-15

Candidato
Ing. Lorenzo Fiorineschi

Tutori
Prof. Ing. Paolo Rissone

Ing. Federico Rotini

Controrelatore
Prof. Ing. Roberto Viganò

Coordinatore del Dottorato
Prof. Ing. Maurizio De Lucia

Anni 2012/2014

To my family

Summary

The research activity described in this thesis came up from an observation made by the author, when performing design activities concerning a new modular system for experimental purposes. He observed that the current methodology does not comprehensively support the designer in early conceptual design activities of new modular products. More specifically he noticed that, without the presence of a preliminary concept, it is currently impossible to systematically face issues related to modularity.

On the base of such an observation, the aims of the work discussed in this thesis arose, which can be shortly resumed in performing a deeper investigation about the preliminary observation and, if confirmed, facing the current methodology lacks. The performed activity was structured as described here in the following, where also a short introduction on obtained results is reported.

The first part has been dedicated to investigations aimed at validating preliminary hypothesis and at improving the knowledge base. Obtained information led to the formulation of more detailed objectives, i.e. the development of a new way for facing modularity and the development of a new conceptual design approach for its structured implementation.

Then, the second part of the PhD concerned the development of preliminary methodological proposals, which were successively merged in a new conceptual design method, equipped with a main graphical tool. The logic of such an overall proposal allows the identification of opportunities to use modularity, and supports the designer in developing modular solutions. Moreover, also some important issues concerning current conceptual design methods were faced.

The last part of the work has been focused on testing activities concerning the developed proposals. These activities were partially conducted concurrently with the development of the approach, and such preliminary tests allowed to refine the target and to focus the attention on some specific issues. On the base of such evaluations, some improvements were made and a final test was performed on a sample of convenience composed by engineering students. The obtained results gave a first validation of the approach and furnished important indications for future developments, aimed at upgrading the proposal towards the application in industrial contexts.

Table of contents

List of figures

List of tables

List of Acronyms

AG: Analysis group.
CAD: Computer Aided Design
CG: Control group.
DSM: Design Structure Matrix.
EMS: Energy, Material, Signal functional model.
ENV: Element-Name-Value model.
FSH: Function Structure Heuristics
GUI: General User Interface
MFD: Modular Function Deployment
MIM: Module Interaction Matrix
NoP: Network of Problems
OTSM: Russian acronym that stands for "Theory of powerful thinking".
PSN: Problems-Solutions Network.
QFD: Quality Function Deployment
SoA: State of Art.
TRIZ: Russian acronym for "Theory for inventive problem solving" (TIPS in English).
VDI: Verein Deutscher Ingenieure (German association of engineers).

List of publications

Main publications (Appendix D for first pages):

Fiorineschi L., Frillici F.S., Cascini G. “Linking TRIZ to Conceptual Design engineering approaches”. 14th ETRIA TRIZ Future Conference 2014 - Losanna (Switzerland), 29–31th October 2014.

Fiorineschi L., Rissone P., Rotini F. “Investigating On The Rise Of Modularity During The Conceptual Design Phase”. International Design Conference - Design 2014, Dubrovnik - Croatia, May 19 - 22, 2014.

Fiorineschi L., Rissone P., Rotini F. “Modularization vs. Innovation”. International Journal of Innovation Science, Vol. 6, no. 1, 2014, ISSN 1757-2223.

Cascini G., Fiorineschi L., Frillici F.S., Rissone P. “Techno-economic classification of contradictions and related strategies of solution” – 13th ETRIA TRIZ Future Conference 2013 - Paris (France), 29–31th October 2013.

L. Fiorineschi, F.S. Frillici, P. Rissone and G. Cascini, Product Architecture definition: evaluating the potentiality of TRIZ tools. 12th ETRIA TRIZ Future Conference 2012 - Lisbon (Portugal), 24–26th October 2012

Other contributions published during the PhD scholarship:

M. Barbari, A. Cavalli, L. Fiorineschi, M. Monti, M. Togni. “Innovative connection in wooden trusses ”. Construction and Building Materials 66, 654-663. Elsevier.

Fiorineschi L., Boscaleri A., Rissone P. Requirements Versus Constraints In Designing Stratospheric Platforms For Multi-User Heavy Payloads: The Large-Scale Polarization Explorer (LSPE) Experiment. ‘21th Symposium on European Rocket and Balloon Programmes and Related Research, , 9-3 June 2013, Thun, Switzerland.

S. Aiola, G. Amico, P. Battaglia, E. Battistelli, A. Baù, P. de Bernardis, M. Bersanelli, A. Boscaleri, F. Cavaliere, A. Coppolecchia, A. Cruciani, F. Cuttaia, A. D' Addabbo, G. D'Alessandro, S. De Gregori, F. Del Torto, M. De Petris, L. Fiorineschi, C. Franceschet, E. Franceschi, M. Gervasi, D. Goldie, A. Gregorio, V. Haynes, N. Krachmalnicoff, L. Lamagna, B. Maffei, D. Maino, S. Masi, A. Mennella, Ng Ming Wah, G. Morgante, F. Nati, L. Pagano, A. Passerini, O. Peverini, F. Piacentini, L. Piccirillo, G. Pisano, S. Ricciardi, P. Rissone, G. Romeo, M. Salatino, M. Sandri, A. Schillaci, L. Stringhetti, A. Tartari, R. Tascone, L. Terenzi, M. Tomasi, E. Tommasi, F. Villa, G. Virone, S. Withington, A. Zacchei, M. Zannoni. The Large-Scale Polarization Explorer (LSPE). SPIE proceedings of the Astronomical Telescopes + Instrumentation 2012 Conference - Ground-based and Airborne Instrumentation for Astronomy IV, Amsterdam 1-6 July 2012, paper #8446-277

1. Introduction

Despite the presence of a plethora of well-acknowledged contributions concerning design methodology, product architecture issues seems to be faced in a separate way by scholars. Indeed, if in the one hand it is possible to find design methods and models, which do not consider product architecture issues comprehensively, on the other hand there are well acknowledged methods aimed only at rearranging architectures of existent products. It implies that, especially for early design phases, a concurrent exploiting of the methodological resources characterizing the two literature branches can be very hard to be realized.

Such an observation arose during a design activity concerning the development of a new system, where the author of this thesis was directly involved. In such a case, the design task was characterized by the explicit request of reaching a modular solution. Indeed, the main objective was the realization of a system characterized by an high reusability and adaptability of the system. In order to fulfill such requirements, the leading idea, generated by experts in the specific sector, was to conceive something constituted by distinct modules. Then, after first researches aimed at finding a methodological help in exploiting the design task, it appeared quite evident that in case of radically new products, no methods or tools existed for directly assisting the designer in "conceiving" modular concepts. This is quite curios if considering the fundamental role of the conceptual design phase for achieving product success, and the number of potential benefits characterizing the adoption of modularity. Indeed, it is well acknowledged that if starting from a poor concept, it can be very hard to achieve product success only by incremental modifications. Moreover, it has been demonstrated by literature that modular architectures may be used in order to save costs related to the different life-cycle phases.

It was on the base of the above mentioned observation that the present research activity started, with the aim of investigating on the actual possibilities to improve the current link between conceptual design and product architecture issues related to modularity. More precisely, the objective of the work can be shortly resumed in two main parts, i.e. performing a deeper investigation about the preliminary observation and, if confirmed, facing the current methodology lacks. For what concerning the second part of the objective, the development of a new methodological proposal is foreseen, which should be capable of supporting the designer in facing modularity issues since earliest concept generation activities.

Therefore, a comprehensive literature analysis has been performed both on engineering design and product architecture focused literature, in order to reach a sufficiently extended knowledge base. Indeed, the gathered information and the improved consciousness concerning the actual potentialities of the existent contributions, formed the base for approaching the successive research activities. The results of such investigation have been split in two distinct parts, i.e. the results concerning the engineering design process (Section 2) and those related to product architecture (Section 3).

Then, on the base of the new amount of available information, it has been investigated the possibility of an hypothetical matching between the current modularity methods and the most acknowledged models for the conceptual design phase. In order to do that, one the most taught conceptual design approach has been considered, i.e. that of Pahl and Beitz, who inspires many other literature contributions, and whose functional model for conceptual design is widely acknowledged and used by scholars. Such a comparison is deeply described in Section 4.

In Section 5, a detailed list of the objectives characterizing the research activity is reported, as a consequence of the observations performed in the previous sections. More precisely, in this section the ideal characteristics of the new methodological proposal are identified and described. Furthermore, in the same section some realistic observations have been reported, concerning the actual resources available for the present work.

Section 6 is dedicated to the description of the development process of the proposal, which can be substantially subdivided into two main groups. The first group concerns the development of a new approach capable of managing modularity since early concept activities. Consequently, the second group concerns the development of a new conceptual design approach capable of implementing the new modularity approach and overcoming most impacting deficiencies of the reference model of Pahl and Beitz.

In order to reach a final version of the proposal, intermediate evaluations have been performed on single parts composing the overall approach. In this way it has been possible to evaluate advantages and lacks of the intermediate models, and then to refine them towards better applicability and potentialities. Such preliminary observations have been performed by means of specific tests in which engineering students were involved. Moreover, also the application of a preliminary version of the approach on an industrial case study has been used as first trial, in order to reach major information for building a first guideline of the method. Successively, such a guideline has been used to perform a final test on a refined version of the approach. A comprehensive description of tests and the related results is reported in Section 7. Section 8 reports discussions about the performed research activity, highlighting how the main objectives have been matched, what is still missing and what are the possible future developments of this work. After that, a conclusive section is reported, resuming the objectives, the performed activity and the obtained results.

A schematic representation of the thesis structure can be observed in Figure 1, where besides the numerical sequence of the sections, the position of the contents in relation to the activity evolution is also represented. Moreover, in the same picture the input/output relationship between sections is represented by means of dashed lines and arrows.

Eventually, references are reported at the end of each section and some specific material for further clarification is reported in Appendices.

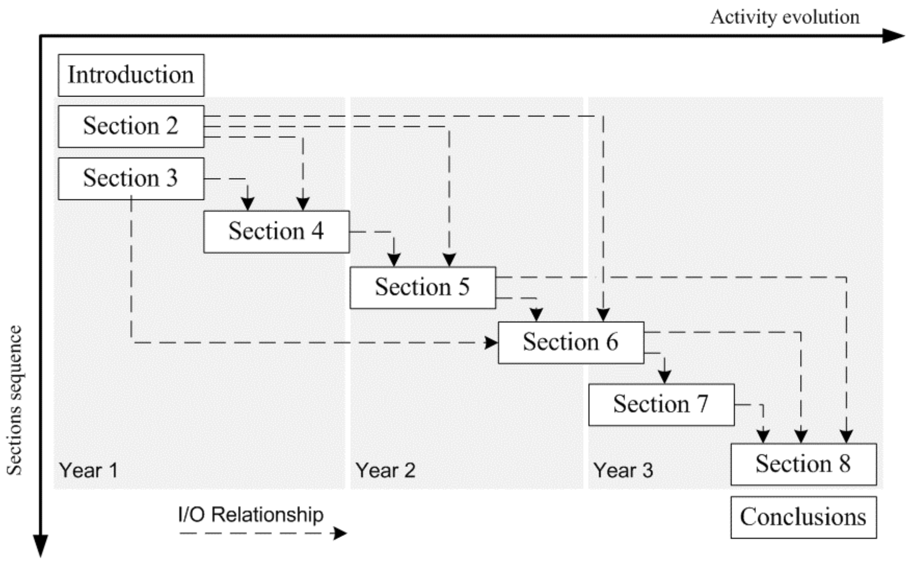


Figure 1. Schematic representation of the thesis structure in relation to the activity evolution.

2. The engineering design process

It is widely acknowledged that the engineering design process plays a fundamental role in the product development, since decisions taken into this phase largely influence the probability of product success. Indeed, it is possible to agree that generally, through the design process, a list of technical and non-technical specifications are translated into technical documents (drawings, manuals, bill of materials, etc.) which guide the subsequent product life-cycle phases. As a matter of the fact, it is possible to find a variety of models and methods in literature, respectively describing the process and guiding the designers in performing it.

Also for what concerns the product specifications, there are many possible definitions, belonging to different engineering branches. Indeed, a comprehensive definition of the requirement list allows to focus the target and then to completely understand the design task, mainly because it is an out and out communication between designers and stakeholders. In other words, a bad or unclear communication of the requirements surely leads to an incomplete matching of the objective, as intended by the stakeholder. Then, also for the scope of this thesis, it is important to clarify what is intended here with the term “requirement”.

Here in this section, a survey of some of the most acknowledged contributions is furnished with the aim of introducing the base of knowledge used for the purposes of this research activity. In particular, some of the most acknowledged definitions of requirements are reported, and a reference one has been highlighted. Then, for what concerns engineering design models and methods, the main literature contributions considered here are introduced and described, keeping particular attentions to the conceptual design phase.

2.1. Product requirements

It is well acknowledged in literature that an engineering design activity is performed with the aim of meeting a certain set of design specifications often called “requirements”, embodying both customer/stakeholder needs and various types of inevitable constraints. On the other hand, an univocal definition of design requirements and constraints is missing (Cascini et al. 2013). Such a lack, for example, may imply complications in the information exchange between the two parts involved in the early stage of a product development, i.e. the Product Planning and the Conceptual Design teams. It often happens that, since product planner has to fulfill customer needs also from a non-technical perspective, the type of information to be processed is too abstract to be directly translated into technical specifications. Besides, also referring the analysis to the engineering design field, many misunderstandings often arise between engineers, e.g. with different backgrounds. Not surprisingly, literature presents a rich variety of definitions, as brief reviewed in the following.

Actually, it is possible to find consensus about the meaning of the term "Functional Requirement" (Glinz 2007), even if this term usually refers to both functional and behavioral aspects of a systems. However, many scholars refer also to another type of requirement, i.e. the "Non-Functional Requirements". For the latter case, the variety of definitions found in literature, implies the impossibility to reach a shared interpretation. For instance, Glinz (2007), reports that those definitions are based on terms like property or characteristic, attribute, quality, constraints and performances.

From the Software Engineering field, a more concise definition of functional and non-functional requirements is given by Paech and Kerlow (2004), who assert that the first type is used to represent "what" the software does, while the second type delineates requirements concerning "how good" the software does something. More generally, Hull et al. (2011) define a Requirement as "a statement that identifies a product or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability".

From the world of Engineering Design, Kamrani and Salhieh (2002) distinguish "Functional Objectives" from two other types of requirements, i.e. the "Operational Functional Requirements" and the "General Functional Requirements". Functional Objectives provide information about the expected functionality of the product. Operational Functional Requirements have been defined as the representation of the set of constraints that the design must possess in order to reach the desired functionality. Instead General Functional Requirements are intended to represent customers' secondary needs.

Cross (2000) includes design specifications concerning performance, size, weight, law and safety under the term "Requirements". Moreover, the same author specifies that statements of objectives and functions should not be considered as performance specifications, due to the lack of an indication of concrete limits.

Roozenburg and Eekels (1991) give a more detailed definition where they define as "Objectives" any statement about the "Goal" of a product development process. Moreover, they identify as "Scaling Objectives" those where it is possible to evaluate alternative solutions in a ranked manner, while "Non-Scaling Objectives" those where solutions can be evaluated substantially only with a binary score. Finally, they define as "requirement" an objective that "any design proposal must necessarily meet", while define as "wishes" all the non-essential objectives.

Maybe one of the most simple and intuitive definition is the one used in the optimization field, i.e. concerning "Objectives" and "Constraints". In fact, an Objective is a goal to which the design activity points, e.g. the maximization or a minimization of a parameter. Conversely, a Constraint is something that needs to be respected in order to make the solution acceptable, e.g. the boundaries between which the final mass of the system has to be included. More generally it is possible to give the following definitions:

- Objective: any goal which has to be reached by means of the design activity, not necessarily expressed by means of reference values.

- Constraint: any limitation, boundary or reference value that restricts the space of possible solutions.

It is possible to observe that these definitions are quite similar to the definitions of Wishes and Demands given by Pahl and Beitz (2007).

2.2. Engineering design models

Observing the plethora of contributions concerning design, it is possible to find many different definitions about terms as methodology, theory, models and methods (see Appendix A.1). However, a comprehensive literature review focused on terms and definitions lies outside the scope of this thesis, where the following reference definitions have been taken into account:

- Design Theory: "… is about how to model and understand design …" (Tomyiama et al. 2009)
- Design Methodology: "…are about how to design or how design should be …" (Tomyiama et al. 2009).
- Design Models: "… which refer to a description or prescription of the morphological form of the design process." (Wynn and Clarkson 2005).
- Design Methods: "… which prescribe systematic procedures to support the stages within a model." (Wynn and Clarkson 2005).

Especially concerning design models, it is possible to find many definitions and related classifications, as reported in Appendix A.1. However, for what concerns the present research activity, it is fundamental to introduce the meaning of the following two groups of literature contributions, i.e. Solution-oriented vs Problem oriented (Wynn and Clarkson 2005):

- Solution-oriented approaches, in which an initial solution is proposed, analyzed and then repeatedly modified as the design space and requirements are explored together.
- Problem-oriented approaches, in which the emphasis is placed upon abstraction and thorough analysis of the problem structure before generating a range of possible solutions.

However, real design problems cannot be solved in a purely problem-oriented fashion. In fact, as reported by Wynn and Clarkson (2005), it is generally recognized that completing a design requires the application of both of these strategies. Nevertheless, it is possible to assert that solution-oriented approaches are strongly influenced by the background forming the base of knowledge of the designer. In this way can be very hard or even impossible to reduce the subjectivity of the results, since preconceptions are necessarily involved in the definition of the starting solution. Problem-oriented models instead, even if are more diffused in academia than in practice, allow to overcome mental barriers by means of abstraction, i.e. generalizing and decomposing the design task in order to face it systematically. Such a peculiarity has been considered fundamental for the scope of this work, then this kind of models have been considered as a reference.

According to Wynn and Clarkson (2005) One of the most acknowledged problem-oriented design models is that of Jones (1963), which is based on the three main stages of analysis, synthesis and evealuation (from Cross 2000):

- Analysis: listing of all design requirements and the reduction of these to a complete set of logically related performance specifications.
- Synthesis: finding possible solutions for each individual performance specification and building up a complete design from these with least possible compromise.
- Evaluation: evaluating the accuracy with which alternative designs fulfil performance requirements for operation, manufacture and sales before the final design is selected.

Wynn and Clarkson (2005) further classify such model into the "abstract approaches" literature category, i.e. the set of contributions "*…which are proposed to describe the design process at a high level of abstraction…*". Moreover, Evbuomwan et al. (1996) put the model into the class of "prescriptive models based on the design process", i.e. those models that tend to furnish indications about the procedural aspects of the design process.

In the same class, Evbuomwan et al. (1996) put many other models among which it is possible to find an extension of the Jones proposal, i.e. that of Watts (1966). Such a contribution is based on the same three processes of analysis-synthesis-evaluation, but they are performed cyclically from an abstract to a more concrete level of the design phases (Figure 2).

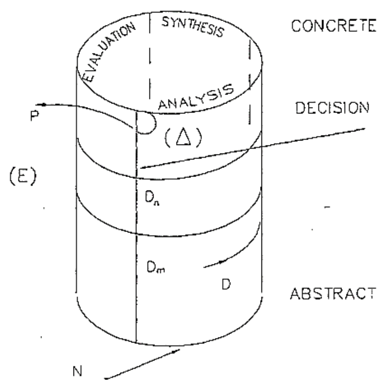


Figure 2. The design model by Watts (from Evbuomwan et al. 1996)

Another model catalogued into the same class by Evbuomwan et al. (1996) is that of Marples (1961), which starts from the formulation of a first design problem to be solved, i.e. the principal node of the so called "Marples tree" (Figure 3). From that node, a first level of sub-problems is derived for each approved solution variant conceived at a high level of abstraction. At this point the process starts again for each of the formulated problems, towards a lower level of abstraction. Then, the Marples tree is characterized by a cyclical repetition of three processes, i.e. analyzing the problem, theorizing solutions and delineating them (Evbuomwan et al. 1996), from a high abstraction level to a more detailed one. Moreover, as represented in Figure 3, the same hierarchical representation of the process can be used as graphical tool for selecting different ways of solving the main problem (or starting node).

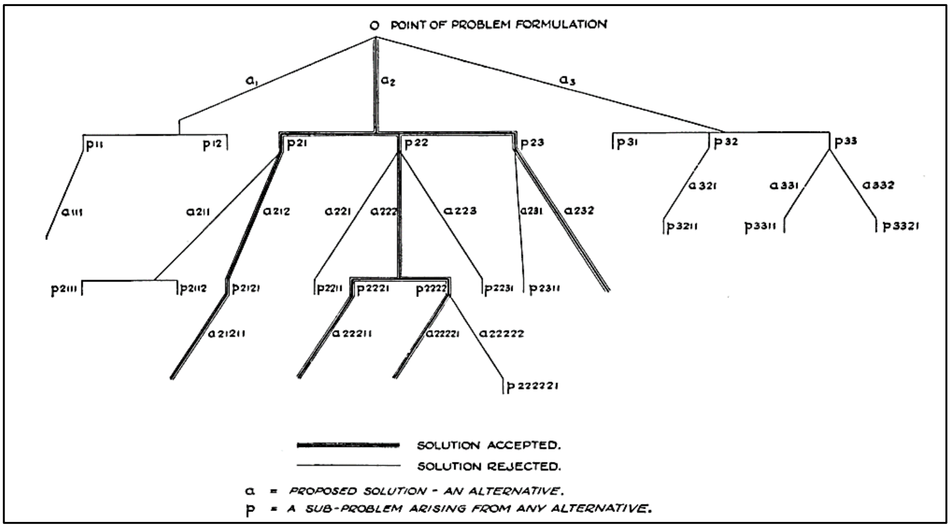


Figure 3. The design process by Marples (from Marples, 1961)

Wynn and Clarkson (2005) when introducing the “procedural approaches” class have considered a more concrete level of abstraction in representing the design process. Two well acknowledged stage-based design models fall under this classification, i.e. that of French (1999) and that of Pahl and Beitz (2007). In the first (Figure 4a), the design process starts by the observation and the analysis of a market need, from which a design problem is stated by a list of product requirements. Then, in the conceptual design stage a set of possible concepts is generated and the most potential of them are selected in order to form the basis for the final product solution. Indeed, the preferred solutions are developed into a more concrete layout during the embodiment stage, and finally all details are developed allowing to proceed with the realization of the production documents.

Concerning the other procedural model considered here, i.e. the Pahl and Beitz one (Figure 4b) it is possible to observe that it shares many commonalities with the French model. Indeed, even in this case the process foresees the formulation of a requirement list, a subsequent conceptual design process followed by the concretization of the system layout (Embodiment design) to be further developed in a detail design phase.

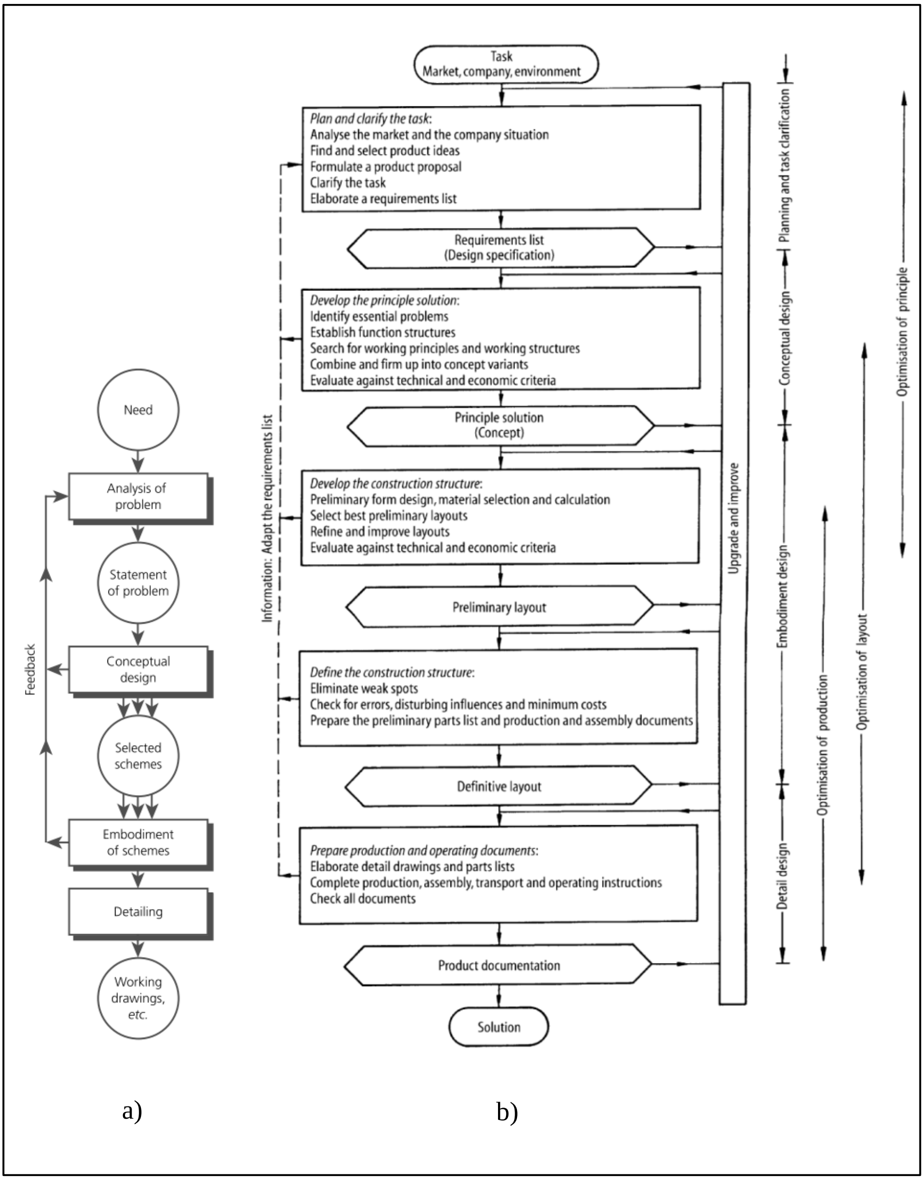


Figure 4. The design processes as conceived by French (a) and Pahl and Beitz (b) (from Cross, 2000)

More precisely, the stages of the Pahl and Beitz model, represented in Figure 4b, can be resumed as follows (Pahl and Beitz 2007):

1. Planning and Task clarification. In this phase, starting from the indications furnished by marketing teams, a list of requirements is obtained in order to focus

the design task. The subsequent design phases should be based on such a list of indications.

2. Conceptual design. Here a first description of the concept, or *principle solution* as called by Pahl and Beitz, is drawn up in order to perform preliminary evaluations. Such a description can be made in many forms, e.g. function structures, flow charts, sketches or rough scale drawings, dependently from the type of information needed to perform a decision. Obviously it is not possible to achieve a total understanding of the solution in this phase, however first evaluations concerning the functionalities and the technological properties characterizing the proposed solution can be extracted.
3. Embodiment design. Starting from the selected concept, the overall layout of the solution is here developed. More precisely, an iterative process is pursued to identify dimension, materials and other technological properties of the system, necessary to assess the financial viability of the project.
4. Detail design. This is the phase where all materials, forms, manufacturing and assembly indications are produced, and all the costs are estimated. Then, "production documentation" is produced, i.e. documents like technical drawings and production specifications.

The Pahl and Beitz model of the conceptual design phase has been considered here as a reference for the development of the research work, and then is described and discussed in detail in the following sub-section.

2.3. The conceptual design phase

Conceptual Design is defined by Pahl and Beitz (2007) as "the part of the design process where (by identifying the essential problems through abstraction, establishing function structures, searching for appropriate working principles and combining these into a working structure) the basic solution path is laid down through the elaboration of a solution principle". However, the model proposed by the same authors is composed by a more detailed series of steps, as represented in Figure 5, which can be resumed as follows:

1. Abstract to identify essential problems.
2. Establish function structures.
3. Search for working principles.
4. Combine working principles.
5. Select suitable combinations.
6. Firm up into principle solution variants.
7. Evaluate variants against technical and economic criteria.

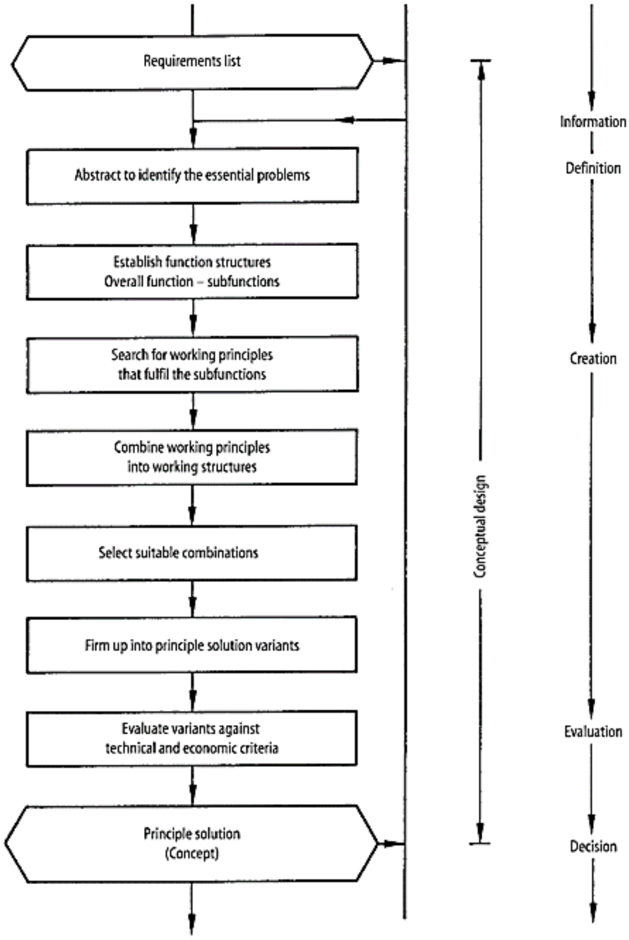


Figure 5. The Pahl and Beitz conceptual design process (Pahl and Beitz 2007)

Some of the most acknowledged models adopted for conceptual design (e.g. those proposed Ullman [2010] or Ulrich and Eppinger [2003]) are based on the central part of the above mentioned process, i.e. the construction of the functional structure of the system as a key step for the achievement of the product concept. Then it is possible to assert that such models have been inspired by the Pahl and Beitz one, especially to its well-known functional model capable of representing the Energy, Material and Signal flows (EMS) (Figure 6). Such a functional model, starting from the outcomes of the task clarification phase, guides the designer towards the definition of the functional structure of the product, from a first general level represented by the overall function of the system, up to lower levels, more and more detailed, constituted by sub-functions.

After that, different possible solutions are identified for the implementation of each function, and a schematic representation of them is listed in a specific matrix. Such a tool is called “morphological chart” (Pahl et al. 2007) or “morphological box” (Heller et al. 2014) (Figure 7), i.e. the main tool of the so-called morphological approach formerly proposed by Zwicky (1966) (as quoted in Heller et al. 2014). In this way the designer has the possibility to try many different combinations among the variety of solutions found for the implementation of each function. A similar approach can be observed in Ulrich and Eppinger (2003), where instead of the morphological chart, the so called “concept combination table” is proposed (Figure 8).

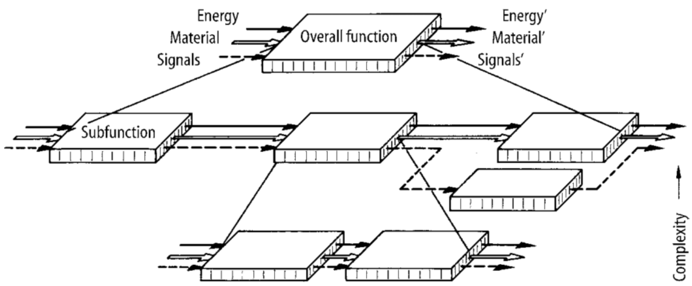


Figure 6. Function modeling by Pahl and Beitz

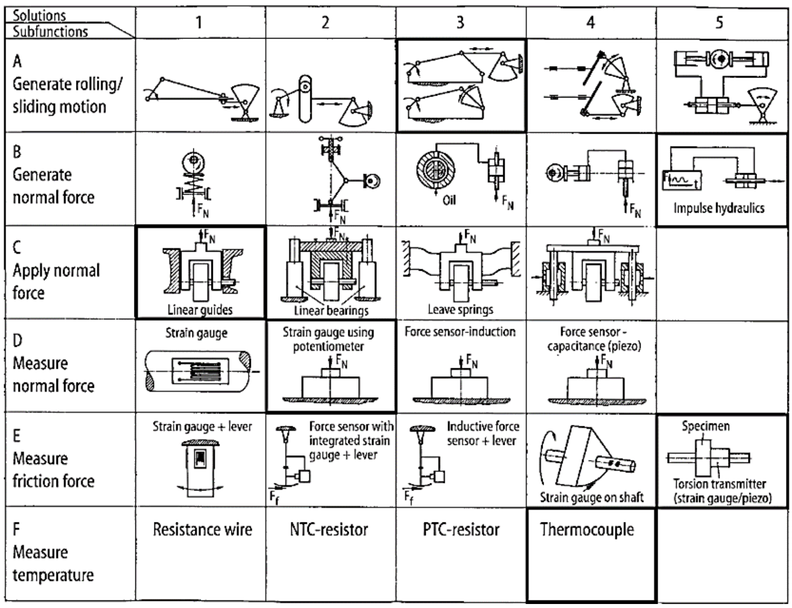


Figure 7. An example of the morphological matrix (from Pahl and Beitz 2007)

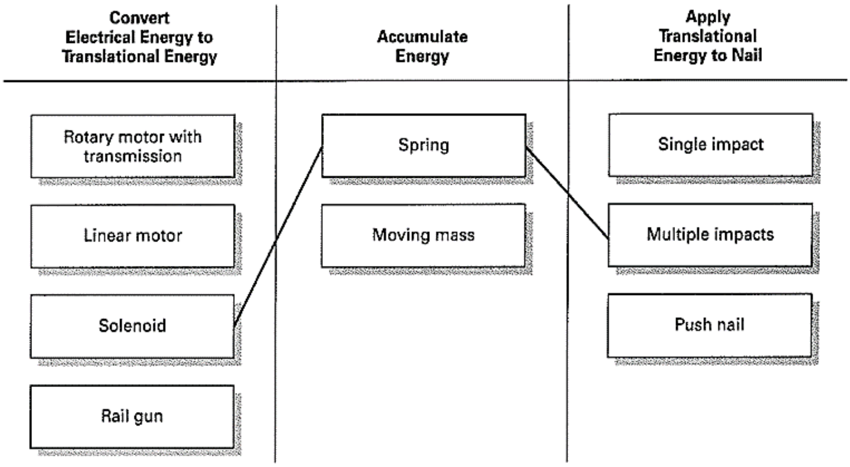


Figure 8. The Concept Combination Table (Ulrich and Eppinger 2003)

After the realization of sketches representing the preferred solution variants, a selection process is performed, by means of specific tools, e.g. the “Concept selection matrix” (Pugh 1991) (Figure 9), “Selection charts” (Pahl and Beitz 2007) or QFD-like matrices (Akao 1990). However, it is not in the scope of this thesis to describe the concept selection phase, since the attention is focused on what concerns the generation and combination activities of the conceptual design process.

	Concepts						
Selection Criteria	**A Master Cylinder**	**B Rubber Brake**	**C Ratchet**	**D (Reference) Plunge Stop**	**E Swash Ring**	**F Lever Set**	**G Dial Screw**
Ease of handling	0	0	–	0	0	–	–
Ease of use	0	–	–	0	0	+	0
Readability of settings	0	0	+	0	+	0	+
Dose metering accuracy	0	0	0	0	–	0	0
Durability	0	0	0	0	0	+	0
Ease of manufacture	+	–	–	0	0	–	0
Portability	+	+	0	0	+	0	0
Sum +'s	2	1	1	0	2	2	1
Sum 0's	5	4	3	7	4	3	5
Sum –'s	0	2	3	0	1	2	1
Net Score	2	–1	–2	0	1	0	0
Rank	1	6	7	3	2	3	3
Continue?	Yes	No	No	Combine	Yes	Combine	Revise

Figure 9. Example of a Concept Selection Matrix (Ulrich and Eppinger 2007).

2.3.1. Considerations about the Pahl and Beitz model

Considering as a reference the widely acknowledged EMS model of Pahl and Beitz, there is the fundamental assumption that design problems can be expressed in terms of solution-independent functions. But as highlighted in Chakrabarti and Bligh (2001), it can results problematic to define such a kind of functions, indeed the function structure of the system depends on the solutions considered for its implementation and vice versa. In confirmation of that, even Pahl and Beitz suggest to imagine a "temporary working structure or a solution" in order to generate the sub-functions needed to perform the functional decomposition process, that leads to alternative function structures of the system. Furthermore, they also recommend to consider such a temporary solution as an intermediate step only, which has not to influence the final solution (Pahl and Beitz 2007, 171). But it is very difficult, or even impossible, to respect such a recommendation during the practical application, because, beyond a certain level of detail, sub-functions are always strictly related to the type of the developed solution. Similarly, also Kroll (2013), among the various criticisms risen about functional decomposition and morphology approaches, asserts that some functions appear only in relation to specific solutions. Then it is possible to infer that before reaching the desired function structure of the product, many different "temporary solutions" have to be considered for evaluating different versions of the overall concept. Moreover, such solutions may belong to different levels of detail and then it can be very hard to realize and evaluate a great number of EMS functional models in a trial and error way. Eventually, it is possible to observe that the composition and evaluation of solutions belonging to different function structures means that different morphological charts, or similar tools, have to be compiled and used, i.e. one for each function structure. But in this way the designer's effort considerably increases, probably generating reluctance in using such an approach.

These observations can be generalized to all design models based on a pure functional decomposition. Tomiyama et al. (2009) include these kind of approaches in what they call the "design methodology" group, and assert that, despite the large diffusion in design education, these methods find a poor industrial application. The same authors assert that one of the reasons of this lack is that these approaches do not emphasize innovative design. Indeed, according to Lenders et al. (2007) (as quoted in Kroll 2013), an excessive use of the functional decomposition limits the designer's freedom, and consequently limits his creativity. In other words, it can be stated that with such an approach the designer limits itself when developing the functional structure of the system, since in order to accomplish the task, he faces the need to focus his attention on some system features even before acquiring all the necessary information. A way to overcome this inconvenient is to consider many different solution variants. However, beyond the above mentioned difficulties, taking into consideration a huge variety of solutions involves the necessity of acquiring a noticeable amount of information, which if not managed properly may lead designers to neglect or even to avoid certain types of solving proposal. Indeed, Sarkar and Chakrabarti (2014) assert that in order to find new solutions in a design space, the designer performs three main types of searches, concerning respectively the solution generation, the problem understanding and the solution evaluation. Therefore, design information are characterized by three different types, i.e. the knowledge to understand design problems, the know-how to solve them and the information needed to accept or discard proposed solutions. But it is acknowledged that different solutions are characterized by different type of knowledge (e.g. technology type, mathematical models, physical laws, etc…) and by a different type of know-how (e.g. engineering methods and specific production technologies), which can be outside the background of the designer or the design team. This is the reason why gathering information and upgrading the requirement list is an activity to be performed continuously during the whole product development, which often triggers the iterative loops characterizing the process. Furthermore, the more the design process moves towards a major concreteness of the outcomes, the more information is needed to take decisions. Keeping track of the need of acquiring knowledge during the design process is then a crucial task when designers try to consider many different potential solutions. Indeed, beyond a certain level of detail, a not negligible amount of the acquired data are often useless for the other solution variants. Unfortunately, conceptual design methods based on functional decomposition do not assist designers in acquiring and managing information “during” the conceptual design phase, even if also Pahl and Beitz (2007, 155) acknowledge that the contents of a requirements list depend on the progress state of the design process.

Eventually, another lack can be ascribed to functional decomposition and morphology methods, i.e. the difficulty in evaluating “how” the design space has been explored. In other words, it can be very hard to understand why a certain functional decomposition strategy has been chosen instead of another one. Maybe also because a shared definition of “what is a function” is currently missing (Eckert et al. 2011). That implies a not negligible level of subjectivity in building function structures, which can result very hard to manage when trying to evaluate the designer activity. Indeed, for functional decomposition and morphology approaches, currently there is no specific tool capable to map the design process in a comprehensive way. Conversely, the mapping of the design moves and the cognitive process of designers is gaining more and more interest for scholars (Becattini et al. 2014), also in terms of coevolution of problems and solutions (Cross and Dorst, 2001).

According to the performed analyses, the drawbacks of the above mentioned conceptual design approaches can be resumed as it follows:

- practical impossibility to avoid prejudices in formulating and decomposing a design problem in terms of functions.
- noticeable efforts in realizing the function structures variants and the related morphological charts.
- difficulties in keeping track of the information gathering activities involved in the decisions taken during the conceptual design process.
- difficulties in managing the design space exploration.

References

Akao, Yoji. 1990. Quality Function Deployment: Integrating Customer Requirements into Product Design. Cambridge: Productivity Press.

Becattini, N., Cascini, G., & Rotini, F. 2014. "An OTSM-TRIZ Based Framework towards the Computer-Aided Identification of Cognitive Processes in Design Protocols". In: Proceedings of the 6th conference on Design Computing and Congnition. Edited by J.S. Gero and H. Sean, 103-122. London: University College.

Cascini G, Fantoni G, Montagna F. Situating Needs and Requirements in the FBS framework, Design Studies, in press, 2013.

Chakrabarti, A., and Bligh, T. P. 2001. "A scheme for functional reasoning in conceptual design." Design Studies 22 (6): 493–517.

Cross N. Engineering design Methods: Strategies for Product Design, 2000, 3rd Ed, Wiley

Cross, N., and Dorst, K. 2001. "Creativity in the design process: co-evolution of problem–solution". Design Studies 22 (5): 425–437. doi:10.1016/S0142-694X(01)00009-6

Eckert, C., Alink, T., and Ruckpaul, A. 2011. "Different notions of function: results from an experiment on the analysis of an existing product." Journal of Engineering Design 22 (11-12): 811-837.

Evbuomwan, N. F. O., Sivaloganathan, S., and Jebb, A. 1996. "A survey of design philosophies, models, methods and systems." Journal of Engineering Manufacture. 210 (4): 301–320.

French, M.J. 1999. Conceptual design for engineers. Berlin Heidelberg: Springer-Verlag.

Glinz M. On Non-Functional Requirements, 2007, 15th IEEE International Requirements Engineering Conference, p. 21–26.

Heller, J. E., Schmid, A., Löwer, M., Feldhusen, J. 2014. "The Dilemma of Morphological Analysis in Product Concept Synthesis – New Approaches for Industry and Academia." In: Proceedings of design 2014, the 13th international design conference. Edited by D. Marjanovic, M. Štorga, N. Pavkovic, and N. Bojcetic, 201–210. Dubrovnik, Croatia, Zagreb: Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb; Glasgow: The Design Society.

Hull E, Ken J, Dick J, Requirement Engineering, 3rd Ed., 2011, Springer-Verlag.

Jones, J. C., and Thornley, D. G. 1962. Conference on design methods. London: Pergamon Press.

Kamrani A K, Salhieh S M. Product Design for Modularity, 2002, 2nd Ed., Kluver Academic Publishers.

Kroll, E. 2013. "Design theory and conceptual design : contrasting functional decomposition and morphology with parameter analysis." Research in Engineering Design 24 (2): 165–183. doi:10.1007/s00163-012-0149-6.

Leenders, R.Th.A.J., Van Engelen, J.M.L., and Kratzer, J. 2007. "Systematic design methods and the creative performance of new product teams: do they contradict or complement each other?" Journal of Product Innovation Management 24 (2): 166–179. doi:10.1111/j.1540-5885.2007.00241.x.

Marples, D. L. 1961. “The decisions of engineering design”. In: IRE Transactions on Engineering Management EM-8 (2): 55-71.

Paech B, Kerlow D. Non-Functional Requirements Engineering - Quality is essential, 2004, Proceedings of the 10th Anniversary International Workshop on Require ments Engineering: Foundation of Software Quality (REFSQ’04), p. 237–250.

Pahl, G., Beitz, W.,” Engineering Design”, 3rd ed., Springer-Verlag, London, 2007.

Pugh, S. 1991. Total Design Integrated Methods for Successful Product Engineering. Reading, Massachusetts: Addison-Wesley Publishing Company.

Roozenburg N F M, Eekels J. Product design: Fundamental and Methods, 1991, Wiley.

Sarkar, P., & Chakrabarti, A. 2014. “Ideas generated in conceptual design and their effects on creativity.” Research in Engineering Design 25 (3): 185–201. doi:10.1007/s00163-014-0173-9.

Tomiyama, T., Gu, P., Jin, Y., Lutters, D., Kind, C., Kimura, F. 2009. “Design methodologies: Industrial and educational applications.” CIRP Annals - Manufacturing Technology 58 (2): 543–565. doi:10.1016/j.cirp.2009.09.003.

Ullman, D. G. 2010. The Mechanical Design Process. 4th ed. New York: Mc Graw Hill.

Ulrich, K.T., and Eppinger, S.D. 2003. Product Design and Development. 4th ed. Irwin: McGraw-Hill.

Watts, R. D. 1966. “The elements of design.” In: The design method, edited by S.A. Gregory, 85-95, London: Butterworths.

Wynn, D., and Clarkson, J. 2005. “Models of designing.” In: Design Process Improvement - A Review of Current Practice, edited by John. Clarkson and Claudia Eckert, 34–59. Springer.

Zwicky, F., "Entdecken, Erfinden, Forschen im Morphologischen Weltbild", München: Droemer-Knaur, 1966.

3. Product architecture

It is acknowledged that a product, or more generally a system, is characterized by its functionalities, the physical principles used for function implementation, and the structure developed to allow the functioning. In particular, concerning the relationship among functions and structures, it is possible to find in literature specific contributions about the so called Product Architecture. One of the most acknowledged definitions of Product Architecture is that of Ulrich (1995) which is based on the distinction between functional and physical elements:

- Functional elements are individual operations and transformations that contribute to the overall performance of the product.
- Physical elements of a product are parts, components or sub-assemblies that implement product's functions.

Moreover, beyond the above mentioned distinction between physical and functional elements, Ulrich considers the specification of the interfaces. Consequently, the definition of Product Architecture takes into account three aspects:

- the arrangement of functional elements;
- the mapping from functional elements to physical components;
- the specification of the interfaces among interacting physical components.

Another basic concept of the definition of Product Architecture, enunciated by Ulrich and Eppinger,(2003) is the "Chunk", i.e. the physical building blocks in which physical elements are organized. Their definition of Product Architecture is the following one: "the scheme by which the functional elements of the product are arranged into physical chunks and by which chunks interact". However, scholars produced other definitions, an excerpt of which is reported in Appendix A.3.

In Pahl and Beitz (2007) Product Architecture is defined as the scheme showing the relationship between function structure and physical configuration of a product; a graphical representation of this definition is shown in Figure 10.

Starting from the mapping between functions and structure, Ulrich (1995) extracts his own definition of modularity and integrality, in fact he considers as modular a product which has a "one-to-one" mapping between functional elements and components (Figure 11), while considers as integral a product which shows one-to-many or many-to-one mapping (Figure 12).

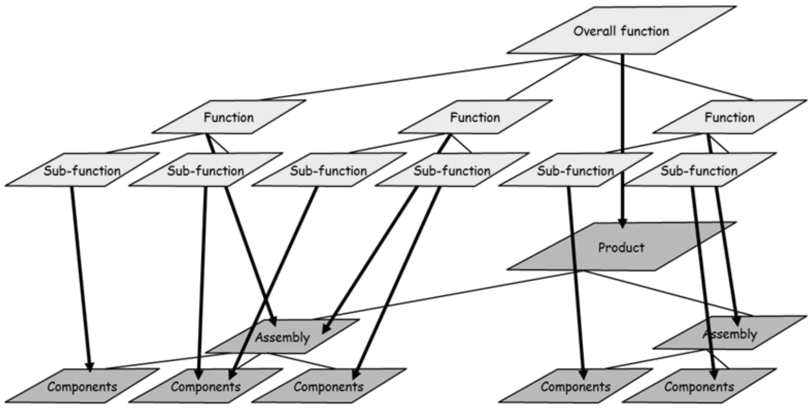


Figure 10. Product Architecture in terms of functional elements and physical elements (Pahl and Beitz 2007).

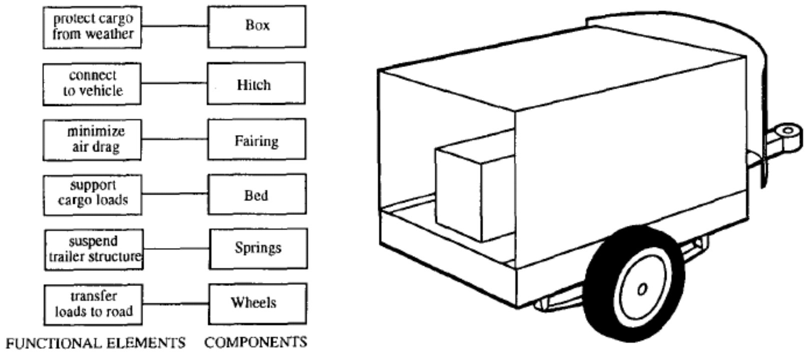


Figure 11. Example of a theoretical modular architecture (Ulrich 1995).

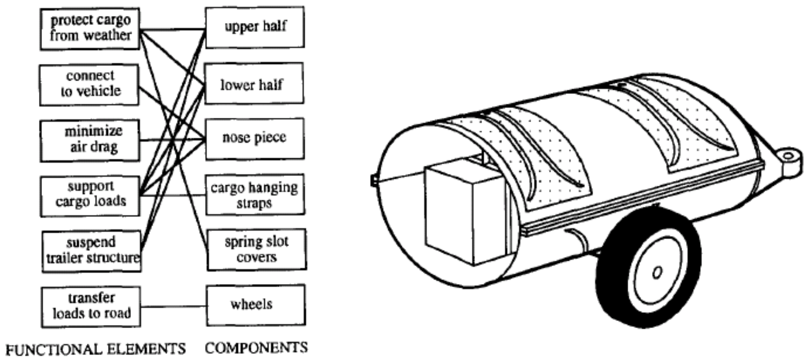


Figure 12. Example of an integral architecture (from Ulrich 1995).

Then, Modularity and Integrality are the two possible types of product architecture, even though in many product it is possible to observe a combination of them, also depending on the considered level of granularity (Chiriac et al. 2011), i.e. the level at which the system has been decomposed (Figure 13).

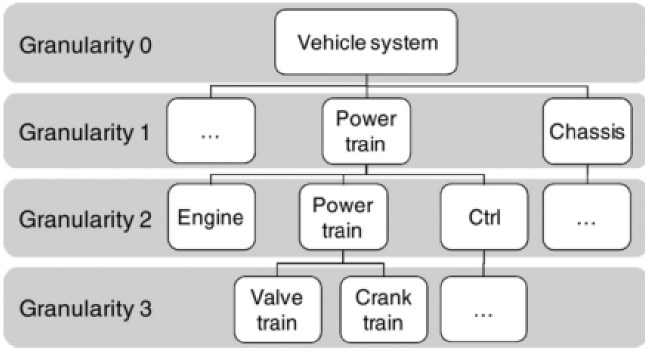


Figure 13. Example of granularity levels (from Chiriach et al. 2011).

The architecture is acknowledged to influence some important product characteristics, and then it is a fundamental aspect to be considered for achieving success. More in particular, the modular type architecture has been widely investigated by scholars. The interest is motivated by the common assumption that “modularization” of products can give rise to a series of benefits. Coming down in a more accurate analysis, some advantages attributed to modularity can be derived from the works available in literature (Gershenson et al. 2003) (Huang 2000) (Ericsson and Erixon 1999), etc. However, modularity is a complex issue, firstly because it is also characterized by drawbacks to be kept under control, secondly because it can be observed and measured in many different, and often, not easy ways (Gershenson et al. 2004). For that reason, the focus of this thesis has been pointed toward modular architectures, and a detailed description of the fundamental characteristics is reported here in the following.

3.1. Modular architecture

Modularity and integrality have, by definition, different implications in product’s performances and costs, but however it is possible to assert that, generally, an integral architecture is a valid solution to reach higher product performances in terms of weight, power or energy consumption. Conversely, modularity may implies negative effects on size, weight and energy efficiency (Whitney 2004). However, a lot of other positive effects have been highlighted for such an architecture type. Furthermore, during the past two decades, a list of well acknowledged types of product modularity have been identified by scholars. Also many definitions can be found in literature, belonging to different engineering domains and based on different perspectives (Salvador 2007). Some definitions may differ when using terms like, module, chunk, component and element. In order to avoid ambiguity and supply a reference for the scope of this work, the following key concepts have been considered, which are based on the consideration that different levels of detail can be identified in a product:

- *System*: Every part or assembly belonging to a determined level of detail may falls under this definition. At the highest level of detail the system corresponds to the product.
- *Component*: With this term, any physical element is identified, intended as single part or assembly, which constitutes the system at the succeeding level of detail.
- *Module*: It is intended here as a particular component connected to the rest of the system by means of decoupled interfaces and which is identifiable with the modularity definitions given in this Section.

Also contributions concerning the definition of modularity measures are present in literature, but for the aim of this work, how much modular is a product is not a fundamental question. Indeed, according to Pahl and Beitz (2007), the goal of modular product development should be to better fulfill customer and/or firm requirements and not merely to have a product more modular than another one. However, a measure of modularity can be useful for comparing different architectures. This purpose anyway justifies the efforts of various scholars during the last decade in order to obtain a shared metrics, many of which are reported in Gershenson et al. (2004).

3.1.1.Modularity benefits

As previously stated, the interest of scholars towards modularity is motivated by the common assumption that, despite some inevitable disadvantages, "modularization" of products can give rise to benefits under many points of view. Many of these benefits have been highlighted by several literature contributions which show the advantages of modularity (Newcomb, Bras, Rosen 1996), (Gu and Sosale 1999), (Huang 2000), (Gershenson, Prasad, Zhang 2003), (De Weck, Hölttä-Otto 2005), (Krause, Eilmus 2011), etc. In Table 1 a list of these benefits is reported and grouped according to the four main product life-cycle phases.

Table 1. Modularity Benefits

Life-Cycle phase	BENEFITS
DESIGN	a) Parallel Development b) Design Reuse c) Design Team management
PRODUCTION	d) Ease of Assembly e) Logistic Optimization for Production/Assembly f) Economy of Scale g) Late Point Differentiation/Customiz. or Postponement
USE/OPERATION	h) Ease of Maintenance/Repair Operations i) Reconfiguration/Flexibility in Use j) Variety k) Customization l) Upgrades/Part Changes
RETIREMENT	m) Material Recycling Facilitation n) Disassembly Time o) Part/component Reuse

That list of benefits have been defined by interpreting and generalizing the contributions currently available in literature, and a detailed explanation of each benefit is reported in the following:

a) With "Parallel Development" is intended the possibility to subdivide the product development task into different and independent development sub-tasks. Indeed, Gu and Sosale (1999) assert that by dividing design and development tasks into more elementary sub-tasks and properly defining the interfaces between the sub-tasks, design teams can carry out them in parallel to reduce product design and development times. Gershenson et al. (2003) confirm this concept, indicating that Modular Design allows the parallel development of sub-tasks. Huang (2000) define this as a modularity benefit called "Decoupling tasks", specifying that the decoupling can result also in the ability to complete tasks in parallel. Furthermore, Krause and Eilmus (2011) state that, since modules are decoupled, it is possible to reduce the complexity of the development tasks and to allow the parallel development of the same modules.
b) With "Design Reuse" is intended the possibility to reuse a part of the design work performed for a specific task, within the development of other products. The definition derives from Gershenson et al. (2003) who state that by the adoption of a modular architecture it is possible to reuse an existing design subjected only to minor changes. Also Hölttä-otto (2005) includes the Design Reuse into the groups of life cycle benefits of modularity.
c) "Design Team Management" identifies the opportunity of reducing communication and coordination efforts into a structured design team. Fixson (2005) reports that the task's structure influence the way in which development teams interact and communicate. More precisely, he asserts that a product with a high level of complexity is usually detrimental for fast product development, since complex process interactions occur. Therefore, he reports that regrouping components into fewer modules can be a possible solution to reduce the development time of a product. Then modularity can be considered as a potential way to reduce communication efforts and, as stated by Krause an Eilmus (2011), to reduce coordination efforts and needs of documentation.
d) The "Ease of Assembly" benefit represents the possibility to reduce assembly and disassembly operation costs. Gu and Sosale (1999) considers the "modularization for assembly" as a way to shorten delivery time of large and complex products. In the work of Gersheson et al. (2003) some literature contributions are reported which state that modular architectures allow to reduce assembly difficulties. In the same paper it is possible to find also citations to a plurality of literature contributions that associate benefits concerning disassembly time to modularity.
e) The "Logistic optimization for Production/Assembly" benefit identifies the opportunity to optimize the production process from a logistical point of view. Gu and Sosale (1999) state that in order to facilitate production processes and expertise, optimize equipment utilization and reduce total assembly time and

costs, modules can be manufactured and/or assembled in most convenient different locations.

f) If various models of a product share identical functions in their functional structures, it is possible to implement them with identical modules. If these common modules are standardized and produced in a large batch size, an increase of efficiency and quality and a reduction of costs can be achieved (Gu and Sosale 1999). Also Huang (2000) states that modules are usually produced in large quantities allowing increasing the economies of scale. The effect of standardization on component costs is well acknowledged, e.g. Ulrich (1995) asserts that usually standard components are less expensive than custom-made ones, primarily because the standard component is produced in higher volumes. On this principle is based the meaning of "Economy of scale" benefit.

g) With "Late Point Differentiation/Customization or Postponement" is intended the delay of assembling of some components in order to optimize delivery costs. Ulrich (1995) states that "the modularity of the product allows variety to be created at final assembly, the last stage of the production process. Some firms are even delaying a portion of the final assembly until the product has moved through the distribution system and it is ready to be shipped to a customer. This strategy has been called "postponement". Fixson (2005) highlights that interfaces between components influence the Postponement and the Late Customization strategies. Therefore, it is possible to infer that decoupled interfaces characterizing a modular architecture can be considered more appropriated for these purposes.

h) The ease of maintenance/repair operations benefit is related to the reduction of complexity and costs of maintenance operations. Newcomb et al. (1996) report that realizing a certain component as a separated module, as opposed to an integral part of the system, aids in both service and product retirement. Previously also Ulrich (1995) confirmed that a possible way to resolve consumption and wear problems can be the adoption of replaceable modules. Gu and Sosale (1999) further confirm what stated above, reporting that "By grouping components into easily disassemble modules, fault analysis and maintenance of the products are more easily facilitated".

i) "Reconfiguration/Flexibility in Use" identifies the possibility to add or modify functionalities of the product. Ulrich (1995) states that a modular architecture allows minimal efforts to bring those required changes in product's functions needed to convert the functionality of the system. Also Gu and Sosale (1999) confirm that by changing the arrangement or adding one or few modules, a system can be reconfigured to perform other functionalities. Other contributions confirming what stated above are reported in the review of Gershenson et al. (2003).

j) With "Variety" is intended the possibility to obtain different product models with a set of standardized parts. Gu and Sosale (1999) assert that a modular product can provide a choice of different product models by simply rearranging few optional modules. Huang (2000) associates the increase of product variety to modular architectures, thanks to the possibility of using different combination of modules. Other confirmations are reported in the review of Gershenson et al. (2003).

k) Similarly to the previous benefit, "Customization" identifies the possibility to obtain different product models, but by means of customized parts. Indeed, it has been stated above that by rearranging modules, customers can have available more product choices (Gu and Sosale 1999). But in addition to that observation, Kamrani and Salhieh (2002) associate the customization property to modular product by considering the possibility of substituting some components with custom-made ones.

l) "Upgrades/Part Changes" represents the possibility of a product to be upgraded by changing components. Huang (2000) reports that upgrades and product/component changes can be easily achieved by the adoption of modular architectures, thanks to the well specified interfaces that allow bringing modifications on modules independently. Also Kamrani and Salhieh (2002) confirm that modular products bring improvements and upgrades by substituting components with more efficient ones.

m) "Material Recycling facilitation" identifies the possibility to reduce the complexity of the procedures needed to recycle some parts of the product. Newcomb et al. (1996) assert that modules can be defined by considering material compatibility for recycling, i.e. grouping compatible components in the same module. Gu and Sosale (1999) also report that "a modular product can facilitate the separation and sorting of different materials for proper recycling or disposal processes".

n) The "Disassembly time" benefit can be explained with the same description of the benefit (d).

o) Eventually, the benefit "Part/Component Reuse" identifies the possibility to recover some parts of the product after the retirement, in order to be re-used in other products. Gu and Sosale (1999) assert that since after the product retirement some components may be still usable, modular design allows grouping them in easily detachable modules in order to facilitate the recovery. Also Krause and Eilmus (2011) consider the Reuse modularity benefit in the "recycling and disposal" life-cycle phase.

3.1.2.Modular characteristics

Well known definitions concerning modules and modularity are based on functional aspects [Stone, Wood, Crawford 2000], [Pahl, Beitz 2007]. However, it is possible to find some definitions of modularity, focused on the physical structure of modules. Table 2 reports a list of them, with a short description and a reference image. The following criterion has been used here to classify the modularity types:

- *Interfaces types of the modules*. Describing the characteristics of the connectivity among the components of the system. The definitions of Slot modularity, Sectional modularity and Bus modularity belong to this class.

- *Interactions within the system*. Describing how the modules are matched together in order to form the system. Swapping, Sharing and Bus modularity fall into this class.
- *Supply type of modules*. Describing the way by which the components of the systems are provided. Fabricated to fit and Mix modularity belong to this group.

A similar attempt to classify modularity types has been done also by [Salvador, Forza, Rungtusanatham 2002] where the considered modularity types are almost the same introduced here, but with some differences in how they are grouped. Strictly for the aim of this work, these differences have been considered as a possible cause of ambiguity.

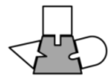
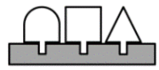
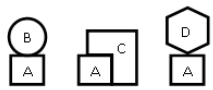
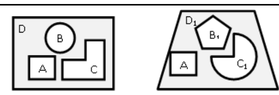
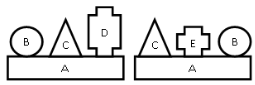
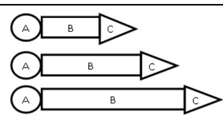
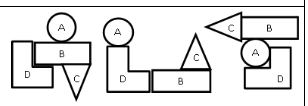
Concerning the first category, (Ulrich, 1995), analogously to (Ulrich and Eppinger 2003), defines three types of modularity on the base of interfaces between interacting components:

- Slot Modularity: all the interfaces between different components are of different type.
- Bus Modularity: it is possible to individuate a common bus that connects other components by the same type of interface.
- Sectional Modularity: all the interfaces between different components are of the same type.

Concerning the second category, Gershenson et al. (2003) as Huang (2000), consider three main types of modularity, based on interactions within a product, which are the three definitions provided by Ulrich and Tung (1991). The first of these definitions is the "Component-swapping" modularity, where two or more components can be interchanged in a module, in order to create product variants. The second definition is called "Component-sharing" modularity, defined as the complementary of Component–Swapping, which consists in a configuration of the product architecture where two or more modules share the same basic component in order to provide product variants. The last typology belonging to this category is another type of Bus modularity, where a module can be matched with any number of basic components. In this case the example used by Huang (2000) in his overview concerns a type of I/O slot module used to plug different computer auxiliary equipment (printer, scanner, plotter, etc.

Ulrich and Tung (1991), as reported in Gershenson et al. (2003) and in Kamrani and Salhieh (2002), define another type of modularity, i.e. what they call the "Fabricate-to-fit" (sometimes called also "Cut-to-Fit") modularity, which substantially consists in combining standard components with customizable components. This type of modularity belongs to the last category of the three listed in this chapter. The second and last type of modularity which belongs to the third category is the so called "Mix" modularity, where a set of standard components can be matched together in order to form a variety of products (Stone, 1997). This is the case of LEGO toy, where many structures and forms can be created by using a limited set of standard blocks.

Table 2. Modular characteristics

Group	Short description	Schematic representation
Interface type	Slot Modularity: all the interfaces between different components are of different type. (Ulrich 1995)	
	Bus Modularity ‘a’: it is possible to individuate a common bus that connects other components by the same type of interface. (Ulrich 1995)	
	Sectional Modularity: all the interfaces between different components are of the same type. (Ulrich 1995)	
Interaction type	Component-swapping:modularity: two or more components can be interchanged in a system in order to create product variants. (Ulrich, Tung 1991)	
	Component-sharing modularity: two or more systems share the same basic component in order to provide product variants. (Ulrich, Tung 1991)	
	Bus modularity ‘b’: where a component can be matched with any number of other basic components. (Ulrich, Tung 1991)	
Supply type	Fabricate-to-fit (sometimes called also “Cut-to-Fit”) modularity: standard components are combined with customizable ones. (Ulrich, Tung 1991)	
	“Mix” modularity, where a set of standard components can be matched together in order to form a variety of products. (Stone 1997)	

3.2. Modularization methods

Several attempts to develop a general method for supporting the designer in reorganizing the product architecture towards modular configurations can be found in literature. However, the Design Structure Matrix (DSM) (Pimmler and Eppinger 1994), (Eppinger and Browning 2012), the Modular Function Deployment (MFD) (Ericsson and Erixon 1999) and the Function Structure Heuristics (FSH) (Stone et al. 2000) can be considered as the representative sample of the most acknowledged methodologies for product modularization (Hölttä-Otto 2005), (Borjesson 2010), (Daniilidis et al. 2011). In order to evaluate the possible interactions between conceptual design and modularization process, it is necessary to understand how these methods works. For that reason, here in the following the logic of each considered method is introduced.

3.2.1.Methods based on Design Structure Matrix

Many authors (Ulrich and Eppinger 2003, Fixson 2003, Sosa et al. 2007, Kamrani and Salhieh 2002) acknowledge Steward (1981) as the first developer of the DSM. Recently he has been defined even as the "grandfather" of the DSM (Eppinger and Browning 2012). This kind of matrix representation was originally used for the analysis of design descriptions (Ulrich and Eppinger 2003), i.e. descriptions which have the purpose of transferring information about the designed artifact. Browning (2001) and, similarly, Kamrani at al. (2002) categorize four versions of DSM that they call Dependency Structure Matrices. The differences with respect to DSM consist in the kinds of dependency that can be represented, as explained in the following:

- Component-Based DSM:
  It is used to represent the system architecture in terms of the functional relationships between the components that form the system. Generally it is used to individuate modules by clustering the matrix. The followed logic consists in the maximization of the interactions between elements belonging to the identified subsystems and the minimization of the interactions between elements belonging to different subsystems.
- Team-Based DSM:
  It represents information flows between different organization elements and the goal is to identify their interfaces and group teams where the flow are most frequent. Sosa et al. (2003) use a team-based DSM to study how the Product Architecture impacts on Design Team interactions.
- Activity-Based DSM:
  It provides an aid in describing input/output relationships between activities that form a project with the aim of optimizing their sequence to cut the cost of the project in terms of required time. Eppinger et al. (1994) use it in order to study interdependences between product development activities
- Parameter-Based DSM:
  Kamrani and Salhieh (2002) state that the fulfillement of design activities is based on the determination of the values associated to parameters that constitute the lowest level of design. This kind of DSM is used to capture the relationship between those parameters, and to rearrange the steps by which parameters are determined.

The first matrix typology, the so called Component-Based DSM, is used to manage the modularization of products thanks to its capability to reorganize the architecture by using matrix manipulation algorithms. This type of DSM (Figure 14) is also called "Product Architecture DSM" by Eppinger and Browning (2012), who define also three additional DSM models, i.e. the "Organization Architecture DSM" the "Process Architecture DSM" and the "Multi-Domain Matrix (MDM)".

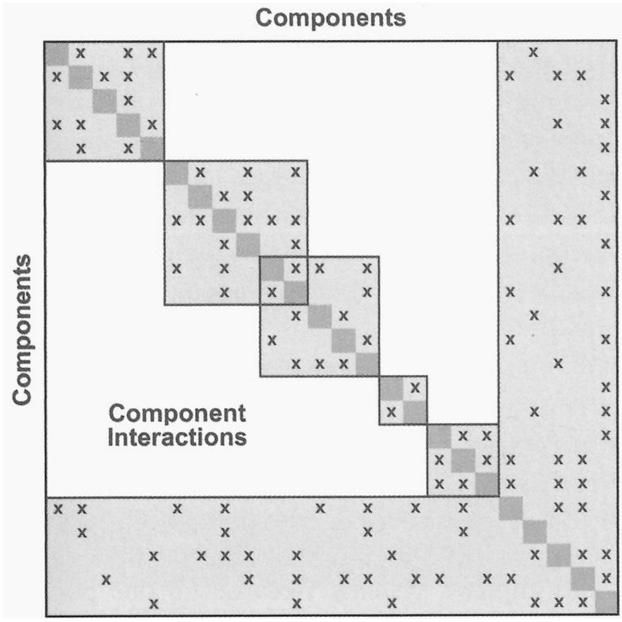


Figure 14. Component based DSM or Product architecture DSM (Eppinger and Browning 2012).

In the Component-Based DSM, the application of clustering algorithms is performed in order to reach groups of components, i.e. modules, in which relations within them are strong, while the relations between them are the more weak as possible.

Kusiak(1998) uses two types of matrices in his modular approach namely the Interaction Matrix and the Suitability Matrix. The first is a component vs. component matrix where values in each box represent the interaction between two elements. The second matrix is still a component vs. component type, where values in each box represent the suitability for two components to be included into a module.

By associating the first matrix with the second one, the so called Modularity Matrix is obtained, where modules and independent components can be identified.

Kusiak (1998) considers a design process as defined by Pahl and Beitz (1999), i.e. subdivided into the following four main phases: Clarification of the task, Conceptual Design, Embodiment design, Detailed Design. For the purpose of this work it is important to note that Kusiak (1998) considers that, even if it is desirable to form modules early in the design process e.g. during the Conceptual Design, the information to identify modules might not be still available at that time. This lack of information hinders the definition of the suitability matrix, thus the identification of module has to be performed during the Detailed Design phase.

Pimmler and Eppinger (1994) develop a special form of DSM in order to manage Spatial, Material, Energy and Information interactions between functions or physical components. More specifically, by using the definition of chunk introduced in Section 3.1, the approach is constituted by three steps:

1) Decomposition of the system into elements. In this step the product is decomposed in its functional and/or physical structures.
2) Documentation of the interactions between elements. Here the interactions between functional and physical elements are listed and subdivided into the four categories: Spatial, Energy, Material, Information. Furthermore a score from 2 (Desired) to -2 (Detrimental) in steps of 1, is associated to each category into the boxes of the matrix.
3) Clustering of the elements into chunks. Here, by means of a clustering algorithms, the rows and columns in the matrix are reordered to cluster the positive elements closer to the diagonal. In the specific example reported in Pimmler and Eppinger (1994), authors use a heuristic swapping algorithm.

Kamrani and Salhieh(2002) use a form of DSM into their structured modularity approach which is constituted by four steps, namely Needs analysis, Product Requirement analysis, Product Concept analysis and finally the Product Concept integration.

In the first step, customer needs are analyzed and managed also by making the use of Quality Function Deployment (QFD) (Akao, 2004) that is a well-known technique for translating customer requirements into technical specifications. The results of step one are used in the second step where functional objectives and constraints are defined. In step three the product is decomposed in functional and physical basic elements. In step four, a component vs. component matrix is realized, in which each element represents the so called “Similarity Index” associated to two elements. Leaving out the details of the method, which is well described in Kamrani and Salhieh (2002) also by means of a practical example, it is relevant to note that they propose a genetic algorithm model for clustering the DSM in order to obtain the modular architecture. Furthermore different solutions differentiated by the number of modules can be ranked by means of the sum of the fitness value, defined as the objective function of the p-median model (reported in Kamrani and Salhieh (2002)).

Tilstra et al. (2009) develop the High Definition Design Structure Matrix (HDDSM) model, in order to capture interactions information to be used for highlighting design improvements. This kind of representation is founded on the definition of a comprehensive standardized basis through which to represent the interactions among the element of the system. The model can be used for quantitative analysis or re-engineering processes of existing products (Tilstra et al. 2012). In the latter case the process is constituted by six overall steps, which start from a reverse engineering activity. Subsequently, parts are assigned to groups and then group-level HDDSMs are merged into the previously created system HDDSM. Finally, the HDDSM is used for the product analysis.

3.2.2.Function Structure Heuristics

The method of Function Structure Heuristics (FSH) has been developed by Robert B. Stone (Stone 1997), where he reports his definition of working heuristics: “a method of examination in which the designer uses a set of steps, empirical in nature, yet proven scientifically valid, to identify modules in a design problem.” The starting point of the method is a general functional structure of the system, that is based on the Pahl and

Beitz model (Pahl and Beitz 1999) as already mentioned in Hölttä-Otto (2005). Even though the method originally adopts a functional basis, as remarked by Stone, it can be employed also without using it. Then, three heuristics have been developed in order to find modules operating on the functional structure of the product (Stone 1997), (Stone et al. 2000). Figure 15 presents how a module is individuated in the function structure through the first heuristic, i.e. Dominant Flow. Stone et al. (2000) have expressed such a heuristic as it follows: "The set of sub-functions which a flow passes through, from entry or initiation of the flow in the system to exit from the system or conversion of the flow within the system, define a module".

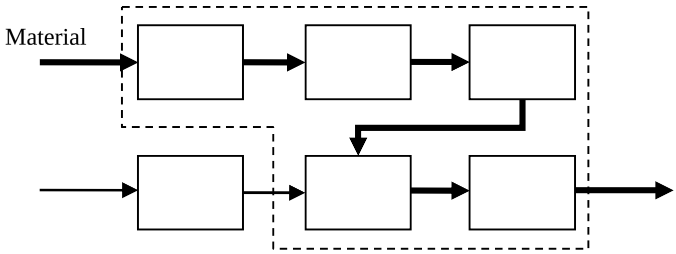


Figure 15. Dominant Flow (Stone 1997, Stone et al. 2000).

The second heuristic is called "Branching Flow", and is defined in this form: "The limbs of a parallel function chain constitute modules. Each of the modules interface with the remainder of the product through the flow at the branch point". A schematic representation of how it works is shown in Figure 16.

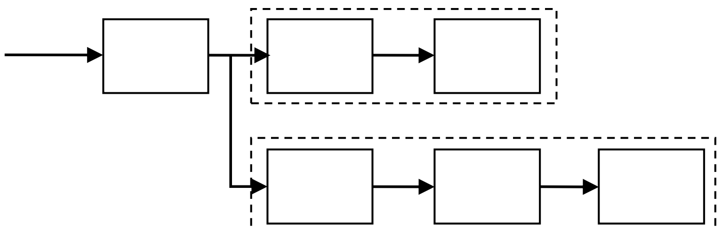


Figure 16. Branching Flow (Stone 1997, Stone et al. 2000)

The last of the three heuristics is the so called "Conversion-Transmission", whose definition allows three possibility to find a module as shown in Figure 17, i.e. "A conversion sub-function or a conversion-transmission pair or proper chain of sub-functions constitutes a module."

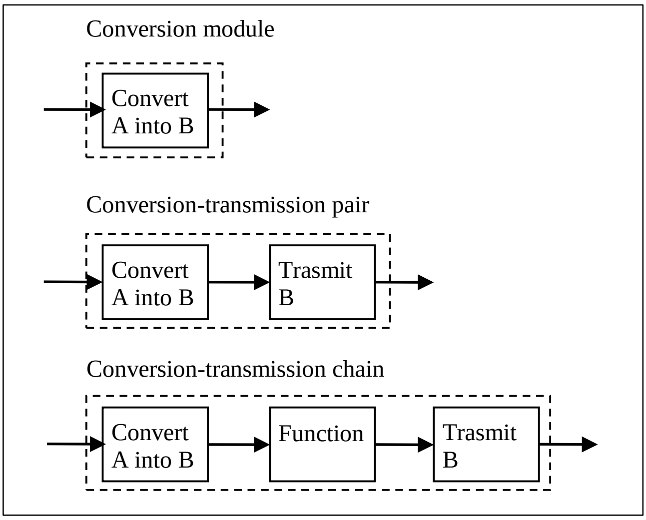


Figure 17. Three possibilities of module identification of the Conversion-Transmission heuristic (Stone 1997, Stone et al. 2000).

According to Hölttä - Otto (2005), this method just brings to modularity suggestions, while the direct intervention of the designer is required in order to choose which module should be implemented. A rule suggested by Stone (1997), is to implement the module with the smallest number of sub-functions. However, many developments of the method have been proposed during the years, e.g., Fixon (2003) reports that efforts have been done in order to include product family considerations, while Gershenson in his review (Gershenson et al. 2004) states that a widening of the work concerned the portfolio architecting.

3.2.3.Modular Function Deployment

The Modular Function Deployment method (MFD) is well described in Erixon and Ericsson (1999), where also some examples are reported in order to help the comprehension. A short description is here reported with the aim to show how the method works and what the most important features are.
Modular Function Deployment is defined by Erixon and Ericsson (1999) as a structured method developed to find the optimal modular product design, taking into considerations company's specific needs. As shown in Figure 18, it is composed by five steps, i.e., "define customer requirements", "select technical solutions", "generate module concepts", "evaluate module concepts" and finally, "improve each module".

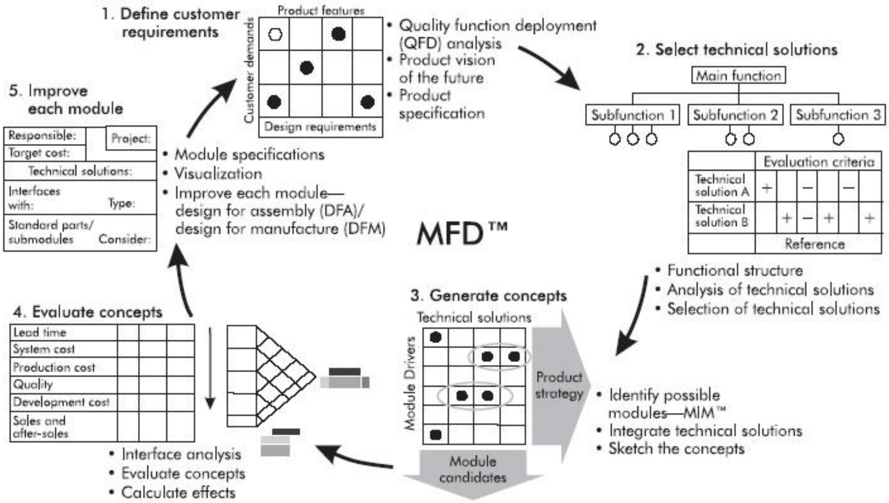


Figure 18. The Modular Function Deployment process (Ericsson and Erixon 1999)

In the first step, a simplified version of QFD is used to link customer needs to product properties; then, in step two, functions are identified and corresponding technical solutions are selected. In step three the previously selected technical solutions are investigated so as to evaluate their possibility to form a module; for this purpose the so called "Module Drivers" and the Module Indication Matrix (MIM) are introduced.Module Drivers are a number of driving forces for modularization, which can be grouped into five classes as showed in Table 3.

The MIM is a matrix in which rows represent module drivers, columns represent technical solutions and each box contains a weight which can assume the following values: 9 for a strong driver, 3 for a medium driver and 1 for a weak driver. The score obtained for technical solutions is used to extract information about modularization. The last two steps are dedicated respectively to the generation of module concepts and to the preparation of technical document to be used for the improvement of modules.

Table 3. Module Drivers (Erixon and Ericsson 1999).

Group	Module Drivers
Product development and design	Carryover
	Technology Evolution
	Planned product changes
Variance	Different Specification
	Styling
Production	Common Unit
	Process and/or organization
Quality	Separate testing
Purchase	Supplier available
After sales	Service and maintenance
	Upgrading
	Recycling

3.2.4.Analysis and comparison of the modularity methods

The performed literature review shows that a lot of efforts have been spent by scholars for the development of design methodologies dedicated to the transformation of an existent architecture in a more modular one. Hence, those methods have been developed for re-engineering purposes, accepting those costs imputable to the necessary iterations into the product life-cycle. Hereinafter, an investigation on the collection of methods previously described is performed with the aim of extracting their positive and negative features. More specifically, many contributions concerning the considered modularity methods have been carefully analyzed, and comments reported by scholars about what is desired for supporting product modularization have been extracted. All of these characteristics have been resumed through seven “descriptors”. Subsequently, the methods have been compared against these descriptors in order to identify lacks also concerning their employment in Original Design tasks.

Building a list of Descriptors for Product Modularity methods

The procedure followed to extract a meaningful list of descriptors to be used in the comparison of the considered methods is shown in Figure 19. Here in the following, a brief explanation is provided of employed approach and obtained descriptors.

Papers related to the three selected methodologies (DSM, MFD, FSH) have been extracted from the literature, and analyzed in order to collect the most important features of the considered methods. That process has been performed primarily by searching in each paper for evaluations, judgments or attributes highlighted by scholars. After that all of these features have been grouped in order to obtain three distinct class, i.e. one group for each method. However, considering the basic differences among the three methods and the variety of the consulted literature, the extracted characteristics were not directly comparable, also because they were expressed in an implicit form through complex sentences. In order to overcome this problem, these characteristics have been generalized in terms of “descriptors” expressing the basic meaning of them.

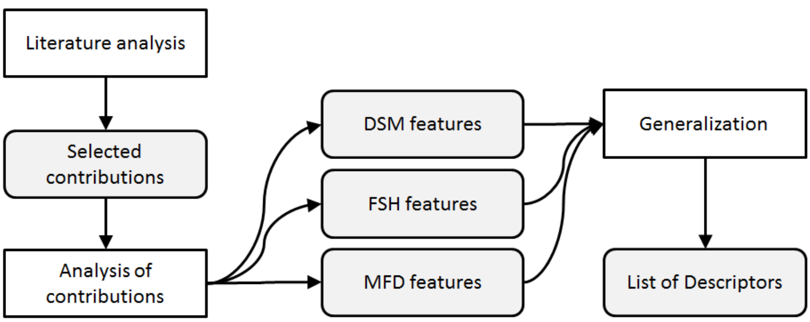


Figure 19. Analysis process followed for the three major modularity methods.

For instance, considering the first descriptor, which is “Management of relations between elements”, it has been obtained by considering literature observations about modularity methods. Indeed, it is possible to find many contributions reporting that DSM methods are used to capture component dependencies (Sosa et al. 2007) and to capture the

relationships between subsystems or components (Tripathy and Eppinger 2011). For those concerning FSH, the method is founded on an Energy-Material-Signal functional model, and then functional relationships are obviously considered (Stone 1997, Stone et al. 2000). In order to abstract the meaning of these observations in one descriptor, it has been considered the definition of product architecture given by Ulrich and Eppinger (2003), where they argue about functional and physical "elements". Therefore, the chosen descriptor is "Management of relations between elements". A similar process has been followed for the definition of the other descriptors.
The descriptor's list is the following:

A. Management of relations between elements
B. Capability to manipulate physical structures
C. Capability to manipulate functional structures
D. Requirements management
E. Multi-domains management
F. Ease of assimilation and usage

A brief explanation of the meaning of the individuated descriptors is reported below:

A. <u>Management of relations between elements</u>:
For the aim of this work, both functional and physical elements (Ulrich and Eppinger 2003) are taken into consideration. This descriptor expresses the capability of a method to allow the mapping of energy, material, information and spatial relationships among the elements. A FALSE value of this descriptor means that the considered method or tool has a poor capability to take into consideration the relations among elements during the design process.

B. <u>Capability to manipulate physical structures</u>:
It is intended as the capability of the method to support systematic modifications of the physical structure of the product so as to fulfill a set of requirements. A TRUE value of this descriptor means that the method allows manipulating the physical structure of the product in order to achieve a certain design objective.

C. <u>Capability to manipulate functional structures</u>:
Similarly to the previous descriptor, it refers to the capability of the method to systematically apply modifications to the functional structure of the product in order to fulfill a set of requirements. A FALSE value of the descriptor means that the considered method cannot operate in terms of functional elements.

D. <u>Requirements management</u>:
Is intended as the property of the method to keep into consideration complex sets of requirements during the design process. A FALSE value means that the considered subject operates independently on the requirements list.

E. <u>Multi-domains management</u>:
Considering the definition of domain reported by Eben et al. (2010), i.e. "a specific perspective of the system", this descriptor is intended as the capability of the method to consider different perspectives of the system during the design process. Examples of domains are physical components, processes, requirements, functions, boundary conditions, etc. (Bauer et al. 2011). A FALSE value means

that up to two or three domains can be considered by the subject, while a TRUE value indicates that an undefined number of domains can be considered.

F. Ease of assimilation and usage:
It is meant as the amount of required knowledge of the user in order to use the tool or the method. A FALSE value stands, e.g., for a method that needs the knowledge of many different tools or a lot of additional information in its steps, while a TRUE value means, e.g., that the user has only to learn method's instructions, without the requirement of other specific high level knowledge. It is different from the "Easy to learn and use" parameter considered in the comparison performed by Borjesson (2010), inasmuch here it is considered also the amount of "additional" knowledge needed by the user in order to follow the various steps of the method itself.

These descriptors have been managed according to the Element Name Value model (ENV) (Cavallucci et al. 2007; Cascini et al. 2009) that is a formalism belonging to the OTSM-TRIZ, i.e. a particular development of TRIZ (Altshuller 1984) originally proposed by G.S. Altshuller himself since 1975 (Khomenko et al. 2007). According to Cavallucci et al. (2007), ENV model is used for various purposes and in particular to allow general description of initial problem situation, to transform it into its model, to analyze the model and, step-by-step, to build the description of a conceptual solution. Furthermore, it can be used in systems description to represent some specific attributes/properties and the state that they can assume. This is shown in the example used by Cavallucci et al. (2007), where by means of ENV model the daily life model used to describe an object, e.g. a tomato, appears as in the following: Object, (Element (E)) named "Tomato" has a set of parameters (attributes), named (N) "Colour", "Shape", etc, and this parameters have a set of associated Values (V), respectively "Red", "Round", etc.

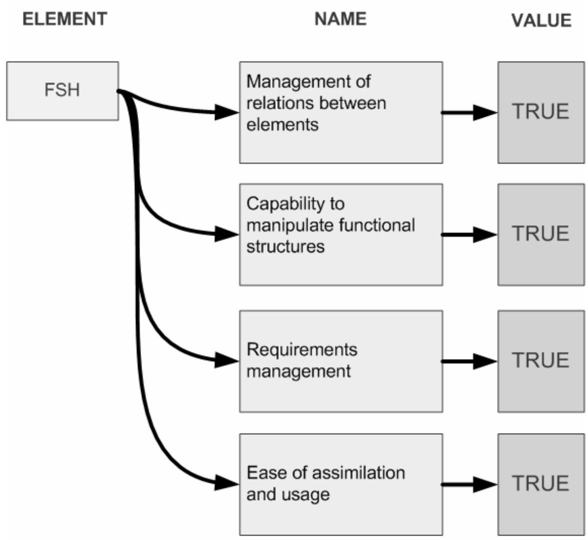


Figure 20. Most relevant features of a Product Modularity method described through an ENV model built upon the literature about FSH.

Concerning the classification of methods according to the descriptors, the "Element" (E) refers to the investigated method, the "Name" (N) refers to the descriptor, i.e. the identified feature/characteristic/property, and the "Value" (V) specifies if the method owns or not the considered characteristic by assigning TRUE or FALSE values respectively. Figure 20 shows an example of the application of the model to FSH method.

In the same picture it is possible to observe that in the specific case all values are TRUE. The reason resides in the fact that from the analysis of the literature concerning that method, it was possible to extract only the characteristics associated and then possessed by the method itself. However, also negative values have been assigned by the author of the thesis, on the base of specific observations. For the other two methods, conversely, some scholars express a deficiency related to particular attributes, so it was possible to assign also the FALSE value. Whereas descriptors don't belong to a specific Element, the corresponding Value has been assigned on the base of author evaluations, reported afterwards.

Analysis of the methods

The descriptors previously identified allow making a comparison between the selected product modularity methods. The results of the comparison are shown in Table 4, where evaluations are expressed only in terms of positive values (TRUE) or negative values (FALSE). The results are hereinafter explained in detail.

Table 4. Results of the comparison.

Descriptors		Modularity Methods		
		DSM	FSH	MFD
A	Management of relations between elements	TRUE	TRUE	FALSE
B	Capability to manipulate physical structures	TRUE	FALSE	TRUE
C	Capability to manipulate functional structures	TRUE	TRUE	TRUE
D	Requirements management	TRUE	TRUE	TRUE
E	Multi-domains management	TRUE	FALSE	TRUE
F	Ease of assimilation and usage	FALSE	TRUE	FALSE

A. Management of relations between elements:
DSM - Since DSM can be used as a project modeling tool that captures the relationships between projects tasks or subsystem/components in a matrix form (Tripathy and Eppinger 2011), it is possible to assert that the characteristics expressed by this descriptor are included. Moreover, as further confirmation, in the matrix representation introduced by Pimmler and Eppinger (1994) it is possible to observe that component, functions, or technical solutions are related in terms of Energy, Material, Information and Spatial relations.

FSH - Since the method (Stone et al. 2000) is based on an Energy-Material-Signal functional model representation (EMS) (Pahl and Beitz 2007), relationships between functional components are considered by definition.

MFD - As deducible from Ericsson and Erixon (1999), where the method is described, there are no tools to look at interfaces or flows between functional components; this is also confirmed by Hölltä-Otto (2005). Indeed, Ericsson and Erixon (1999) show that in the step four there is an instrument used to evaluate modules interfaces, but, according to Hölltä-Otto (2005), it is not detailed enough.

B. Capability to manipulate physical structures:

DSM – One particular form of DSM, i.e. the component-based DSM (Browning 2001; Kamrani and Salhieh 2002) in which components are placed on the row and columns headers of the matrix, is able to reorganize the architecture by using matrix manipulation algorithms. Furthermore, as reported above, relationships between components can be managed, thus it is possible to assert that the DSM tool allows the systematic manipulation of the physical structure.

FSH - As it is observable in the work of Stone et al. (2000), the method allows directly manipulating only the functional structure of the product, so the score for this descriptor is FALSE.

MFD - Since the method exposed in the work of Ericsson and Erixon (1999) operates on technical solutions in order to choose which of them can form a module, it is possible to assess that it can directly manipulate the physical structure of the product.

C. Capability to manipulate functional structures:

DSM – The Function Based DSM (Hölltä-Otto 2005), analogously to the Component-Based DSM, is a matrix representation with functions placed on the row and column headers, and by clustering algorithms they can be grouped together. As stated before for descriptor B, relationships between functions can be managed, so also in this case, systematic manipulation is allowed.

FSH - According to Hölltä-Otto (2005), this method is based on Pahl and Beitz's function structure (Pahl and Beitz 2007), and then it can directly manipulate function structures by definition. Furthermore, also Börjesson (2010) states that heuristics are used to represent function's data types.

MFD - Since in step two of the method (Ericsson and Erixon 1999), i.e. the selection of technical solutions, there is a direct dependency between functions and solutions, it is possible to claim that MFD can manipulate the functional structure of the product.

D. Requirements management:

DSM - In the method proposed by Kamrani and Salhieh (2002) not only the relations between function and components are included, but also the effect of these relations on the satisfaction of customer requirement.

FSH - Stone et al. (2000) in their work take into account customer needs in to choose which module has to be implemented. Gershenson et al. (2004) confirm what stated above and furthermore they report that also expansions of the method of Stone et al. (2000), done by other authors, include customer needs analysis.

MFD - Step one of the method (Ericsson and Erixon 1999) is exactly the definition of customer requirements, by means of which, using a QFD matrix, the designer can rank product attributes.

E. Multi-domains management:
DSM - Bauer et al. (2011) use a Multiple-Domain Matrix (MDM) approach in order to manage the problem of modularity considering different domains. The DSM is a fundamental part of this method, which can be considered as an extension of classical DSM methods.

FSH - Considering for example the method proposed by Stone et al. (2000), it operates into two domains, i.e. the functional domain and the customer needs domain. However nothing has been found in literature to assess that the FSH method is capable to allow the management of further domains, as they are defined by Bauer et al. (2011) and by Eben et al. (2010).

MFD - As reported by Daniilidis et al. (2011), MFD can consider design, manufacturing, use and recycling issues, and furthermore, using the definition of domains given by Eben et al. (2010), it is possible to infer that the method is suitable to manage the system at least from functional, physical, requirements and attributes perspectives.

F. Ease of assimilation and usage:
DSM - The method needs the help of a computer to implement complex algorithms in order to cluster the matrix. The choice of the clustering algorithm is not trivial for a practitioner since a lot of different types are available. Hölltä-Otto (2005) reports that it can suggest overlapping or functionally infeasible solutions, then it needs a continuous control by experts. The need of a direct involvement of experts in order to extract most of the DSM models has been highlighted also by Eppinger and Browning (2012). These reasons have been considered sufficient to assign the FALSE value in Table 5.

FSH - According to Hölltä-Otto (2005), the FSH method is characterized by a low level of complexity of usage inasmuch, once heuristics have been assimilated by the user, no other tools are needed, except for a pencil and a paper sheet.

MFD - According to Hölltä-Otto (2005), the method can be laborious. Furthermore there are several tools to be used during the five steps of the method. Then, taking the FSH methodology as a reference of easiness, the value assigned in this case is FALSE.

Due to the fact that most of the DSM modular design methods are used to group physical components into modules by clustering the matrix, according to (Daniilidis et al. 2011) they cannot be used in early phases of design, because of the lack of information about the structure. Advantages of this methodology are related to the possibility to manage systems characterized by high number of components thanks to the well-known capability of matrices to be implemented in automatic calculations.

FSH method is based on function structure definition of Pahl and Beitz (2007), and it uses different heuristics in order to form modules from product’s function structure.

This characteristic allows using the method in early design phases, furthermore the heuristic approach enhance its ease of usage, but, for example, a lack can be observed concerning the “E” descriptor, i.e. managing multi domain product requirements.

Finally, MFD method is a powerful tool that allows considering a variety of modularization forces, namely Module Drivers, during the development of the modular product. Moreover, customer requirements and company profile are taken into account, but it is not equipped with any instrument or guideline to be used in early phases of concept definition of a new product, e.g. the choice of working principles to adopt.

References

Akao Y. (1990) Quality Function Deployment Integrating Customer Requirements into Product Design. Productivity Press, Cambridge, MA.

Altshuller GS(1984). Creativity as an exact science: The Theory of the Solution of Inventive Problems. Gordon and Breach,New York.

Bauer W, Daniilidis C, Lindemann U (2011) Approach for a Modularization Driven System Definition using Multiple Domains. 13th International Dependency and Structure Modeling Conference, DSM'11,September 14 – 15 2011 Cambridge.

Borjesson F. (2010) A Systematic Qualitative Comparison of Five Approaches to Modularity. International Design Conference, Dubrovnik - Croatia, pp 147-156.

Browning T.R. (2001) Applying the Design Structure Matrix to System Decomposition and Integration Problems A Review and New Directions. IEEE Engineering Management Review, Rev. 48 , 292-306.

Cascini G,Frillici F S, Jangtschi J, Kaikov I, Khomenko N (2009) TRIZ, Theory of Inventive Problem Solving, Improve your problem solving skills. Education and Culture DG, Lifelong Learning Programme, TETRIS (Teaching TRIZ at School).

Cavallucci D and Khomenko N(2007)From TRIZ to OTSM-TRIZ. Addressing complexity challenges in inventive design. International Journal of Product Development, Vol. 4, 1-2/2007, Pages 4-21.

Chiriac N., Holtta-Otto K., Lysy D., Suk Suh (2011) E. Level of Modularity and Different Levels of System Granularity. Journal of Mechanical Design, vol. 133, no. 10.

Daniilidis C., Enßlin V., Eben K., Lindemann U. (2011) A Classification Framework for Product Modularization Methods. International Conference on Engineering Design, ICED11

De Weck, O., Hölttä-Otto, K.," Tradeoff Between modularity and performance for engineered systems and product", International Conference on Engineering Design ICED 05, Melbourne, 2005.

Duffy A.H.B. (1993) Andreasen M.M., MacCallum K.C., Reijers L.N. Design co-ordination for concurrent engineering. Journal of Engineering Design, pp 251–261.

Eben KG, Daniilidis C, Lindemann U(2010) Interrelating and Prioritising Requirements on Multiple Hierarchy Levels. International Design Conference - Design 2010, Dubrovnik - Croatia, May 17 - 20.

Eppinger S.D., Browning T.R. (2012) Design Structure Matrix Methods and Applications. MIT Press.

Eppinger S.D., Whitney D.E., Smith R.P., Gebala D.A. (1994) A Model-Based Method for Organizing Tasks in Product Development. Research in Engineering Design, vol.6, no. 1, pp. 1-13.

Ericsson, A., Erixon, G., "Controlling Design Variants. Modular Product Platform", Society of Manufacturing Engineers, Michigan, 1999.

Fixson S.K. (2003) The Multiple Faces of Modularity: A Literature Analysis of Product Concept for Assembled Hardware Products. Technical Report 03-05, Industrial and Operation Engineering, University of Michigan

Fixson, S.K., 2005. Product architecture assessment: a tool to link product, process, and supply chain design decisions. Journal of Operations Management, Vol. 23, no. 3–4, pp. 345-369.

Gershenson, J.K., Prasad, G.J., Zhang, Y., "Product modularity: Definitions and benefit", Journal of Engineering Design, Vol. 14:3, 2003, pp. 295-313.

Gershenson, J.K., Prasad, G.J., Zhang, Y., "Product modularity: measures and design methods", Journal of Engineering Design, Vol. 15:1, 2004, pp. 33-51.

Gu, P., Sosale, S.,"Product modularization for life cycle engineering", Robotic and Computer Integrated Manufacturing, vol. 15, 1999.

Hölttä-Otto, K., "Modular Product Platform Design", Doctoral Dissertation, Helsinki University of Technology. , 2005.

Huang, C.C., "Overview of Modular Product Development", Proc. of the Natl. Sci. Counc., Republic of China 2000, vol. 24, no. 3, pp. 149-165.

Kamrani, A.K. and Salhieh, S. M., 2002. Product Design for Modularity, 2nd ed. Kluwer Academic Publishers.

Khomenko N,De Guio R, Lelait L, Kaikov I (2007) A framework for OTSM TRIZ-based computer support to be used in complex problem management. International Journal of Computer Application in Technology.Vol 30, Issue ½.

Krause, D., Eilmus, S., "A Methodical Approach For Developing Modular Product Families", International Conference on Engineering Design, ICED 11 Stockholm, 2011.

Kusiak A., Huang C.C. (1998) Modularity in design of products and systems. IEEE Transactions on Systems, Man, and Cybernetics - Part A Systems and Humans, vol. 28, no. 1, pp. 66-77

Newcomb, P.J., Bras, B., Rosen, D.W., "Implications Of Modularity On Product Design For The Life Cycle", ASME Design Engineering Technical Conferences and Computers in Engineering Conference, Irvine, California, 1996, pp. 1–12.

Pahl, G., Beitz, W.," Engineering Design", 3rd ed., Springer-Verlag, London, 2007.

Pimmler T.U., Eppinger S.D. (1994) Integration analysis of product decompositions. Proc. of ASME Design Engineering Technical Conferences, 6th Int. Conf. on Des. Theory and Methodology Conference.

Salvador F., Forza C., and Rungtusanatham M., "Modularity, product variety, production volume, and component sourcing: theorizing beyond generic prescriptions," Journal of Operation Management., vol. 20, no. 5, 2002, pp. 549–575

Salvador, F., "Toward a Product System Modularity Construct : Literature Review and Reconceptualization", IEEE Trans. Eng. Manag., vol. 54, no. 2, 2007, pp. 219–240.

Sosa M. E., Eppinger S. D., Rowles C. M. (2003) Identifying Modular and Integrative Systems and Their Impact on Design Team Interactions. Journal of Mechanical Design, vol. 125, no. 2, pp. 240-252.

Sosa M.E., Eppinger S.D., Rowles C.M. (2007) A Network Approach to Define Modularity of Components in Complex Products. Journal of Mechanical Design, Vol. 129, Issue 11.

Steward D.V. (1981) The design structure system A method for managing the design of complex systems. IEEE Transactions on Engineering Management, Vol. 28, pp. 71–74.

Stone, R.B. (1997) Towards a Theory of Modular Design. Doctoral Dissertation. Faculty of the Graduate School of the University of Texas at Austin.

Stone, R.B., Wood, K.L., Crawford, R.H.,"A heuristic method for identifying modules for product architectures", Design Studies, Vol. 21, Issue 1, 2000, Pages 5-31.

Tilstra A.H., Seepersad C.C, Wood K.L. (2009) Analysis of Product Flexibility for Future Evolution Based on Design Guidelines and High Definition Design Structure Matrix. Proceedings of IDETC, CIE 2009, pp. 1–14.

Tilstra A.H., Seepersad C.C, Wood K.L. (2012) A High-Definition Design Structure Matrix (HDDSM) for the quantitative assessment of product architecture. Journal of Engineering Design, vol. 23, pp. 767–789.

Tripathy A, Eppinger SD (2011) Organizing Global Product Development for Complex Engineered Systems. IEEE Transactions on Engineering Management, Vol. 58, no. 3, August 2011.

Ulrich K. (1995) The role of product architecture in the manufacturing firm. Research Policy, Vol. 24, no. 3, pp. 419-440.

Ulrich, K., Tung, K.,"Fundamentals of product modularity", Proceedings of the 1991 ASME Design Engineering Technical Conferences—Conference on Design/Manufacture Integration, Miami, FL, 1991.

Ulrich, K.T., Eppinger, S.D. , "Product Design and Development", 4th ed. McGraw-Hill, Irwin, 2003.

Whitney D.E., 2004. Physical limits to modularity. Working paper, ESD-WP-2003-01.03-ESD, Massachusetts Institute of Technology, Engineering Systems Division.

4. Conceptual Design vs Modularization Methods

In the Pahl and Beitz design process presented in Section 2, the possibility to systematically form modules is not directly contemplated. However, the same authors foresee the possibility to create modules, by operating on the functional structure of the product (Pahl and Beitz 2007). For that purpose, they provide some definitions from a functional point of view. More precisely, they considers the following types of modules:

- Basic modules: which are intended to implement the so-called “basic functions”, i.e. those fundamental functions which can fulfill the overall function.
- Auxiliary modules: implementing the “auxiliary functions”, i.e. those functions that indirectly contribute to the fulfillment of the overall one.
- Special modules: implementing complementary functions, as for accessories, optional parts and so forth.
- Adaptive modules: which are necessary for adapt the system to boundary conditions, i.e. implementing adaptive functions.
- Non-modules: implementing custom functionalities not considered in the development of the modular product.

Considering such a formulation, at least, a functional model of the system has to be developed in order to identify the above mentioned modules. Indeed, Pahl and Beitz suggest to first realize a functional model of the system, and then to identify the various types of functions, in order to try to group them into specific modules. However, such an approach has not been sufficiently supported by industrial applications.

The possibility to work on modularity early in the design process is also considered in VDI 2221 (Figure 21), where it is possible to observe that modularity considerations can be made after the identification and the combination of the solution principles. But even in this case, a comprehensive support in module definition is missing.

However, considering the above mentioned contributions, widely acknowledged as milestones of design methodology, it is possible to infer that modularity can be applied, at least, only after a comprehensive functional modeling of the solution.

In order to validate such a statement, some considerations need to be expressed on the most acknowledged modularization methods presented in Section 3. In fact, thanks to the results of the comparison performed in Section 3, some analogies can be observed between modularity methods and conceptual design. Indeed, starting from descriptor “A”, it is possible to observe that the management of relations between functional components, between physical components, and among components of both types plays a key role for modularity methods. This is in analogy with the conceptual design, which also contemplates both the functional modeling and the management of working structures.

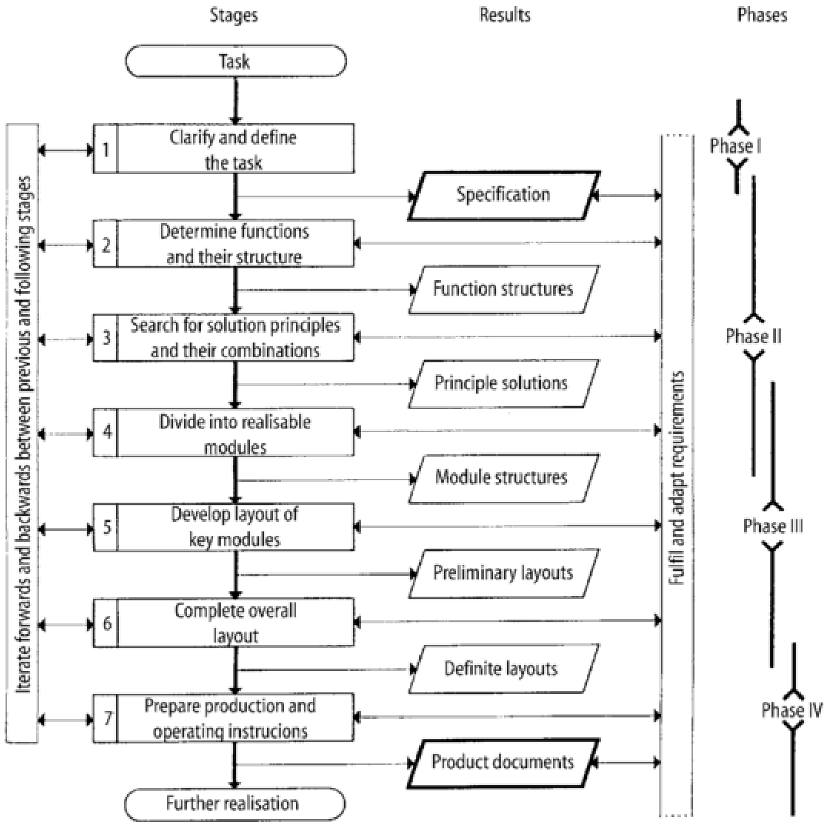


Figure 21. General approach to design for VDI 2221 (From Pahl and Beitz 2007).

What asserted above can be clearly repeated for descriptors “B” and “C” introduced in the previous section, inasmuch the capability to manage functional and physical structures pertains to the current state of the art of modularity methods. From the other point of view, referring on the definition reported in Section 2, it is obvious that a conceptual design method must allows handling both the structures, and then also in this case the analogy exists.

Requirements management is fundamental in Concept Design, because any new product and then, any new idea, is thought in order to fulfill a set of product requirements derived from a set of customer needs, together with a set of constraints of various nature. Results concerning the descriptor “D”, show that the considered modularity methods are also able to manage requirements.

Results related to the management of multiple perspectives of the system, i.e. the descriptor “E”, demonstrate that this characteristic appears in two of the three considered methods. Product Architecture problems are composed by multi-domain requirements and constraints, so a modularization method has provide this functionality. Multi-domain management is not specifically considered during conceptual design, but this characteristic can be considered inherent, since developing a concept means substantially trying to fulfill a set of requirements, internal and/or external to the firm, complying with the constraints imposed from different perspectives, e.g. production process, human resources etc.

However, despite these analogies, conceptual design and modular design seems to be two distinct worlds, and it is not trivial to find a way to reach a comprehensive integration of them. The main reason seems to be the need of modularity methods to operate on a pre-existent model of the product or, at least, a part of it. In order to validate such a statement, here in the following, some reflections about the three reference approaches for modularity are reported, focusing the attention on their potentialities to be integrated in the reference conceptual design approach considered in this thesis.

4.1. Methods based on Design Structure Matrix

Daniilidis et al. (2011) assert that since the DSM-based modularization methods use a component-based analysis of the Product Architecture, they cannot be used to manage modularity during the early concept design phase, due to the lack of information regarding the structure. In fact, all the contributions quoted in Section 3 concerning DSM based methods, start from the decomposition of an existent system, in order to identify elements and then to form the starting matrices. It is evident that a reference product is required, then in case of original design, these methods can be used only after a detailed concept generation aimed at creating system elements. It corresponds to an iterative procedure, where no information about modularity are available during the early concept creation. Similarly, also in case of the upgrading of an existent product, the DSM based methods can only give support in the component rearrangement, but only after the definition of the new product concept. However, despite the impossibility to apparent use the DSM approach to manage modularity during early concept generation, DSM clustering can be useful to guide the designer in the identification of some of the modularity types (Kusiak and Huang 1998) listed by Salvador et al.(2002) (Figure 22). In other words, once a first draft of the concept has been defined, and interaction between components are identified, it may be possible to choose a specific module interaction type and then to cluster the matrix in that sense. However, during the early concept generation it is impossible for the designer to know if the conceived concept is capable of reaching the desired modularity.

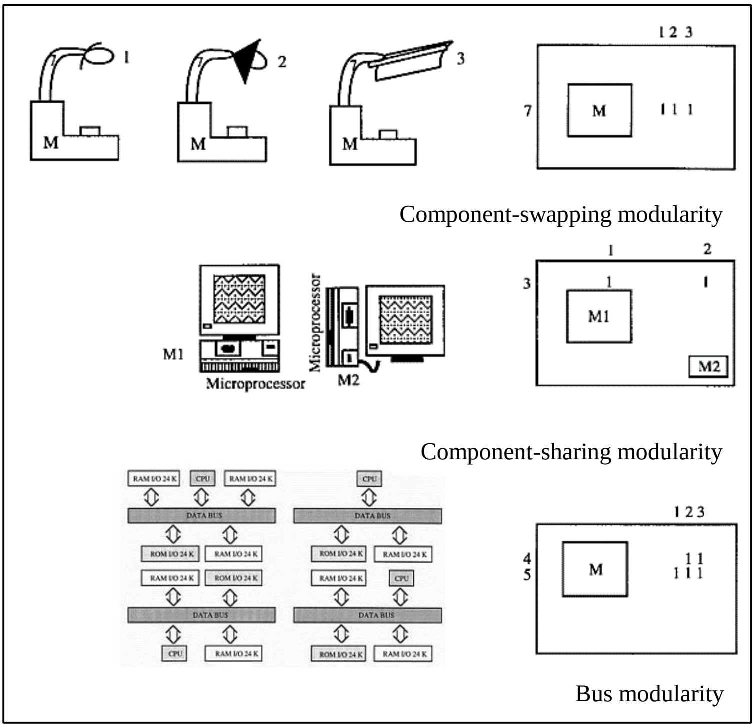


Figure 22. Different module interaction types identified by means of DSM clustering (Kusiak and Huang 1998).

4.2. Function Structure Heuristics

The core of the FSH method is based on the achievement of a comprehensive functional model of the product in order to apply the three heuristics and to find module concepts. The functional decompositions starts with the definition of a black box representing the overall function of the product, together with the energy, material and signal fluxes defined by the analysis of the requirements. However, identifying the fluxes implies the identification of the physical principles on which the functioning of the product is based. For instance, the presence of electrical energy fluxes implies to foresee that something in the system will use physical principles based on electricity. In order to obtain the subfunctions, Stone et al. (2000) suggest that each flow must be examined in deep, from the input to the output of the system, searching for every operation on the flow itself. But it can be very hard to foresee all the operations which characterize the product on every level of detail, especially only from a functional point of view. In fact, depending on the type of product and the available requirements, information about the physical principles considered for the solution have to be known in order to further decompose a function. For example, in Figure 23 two different types of orange juicer have been reported. Since they are both manually operated, plastic made and easy to use, just for example it is possible to consider that the requirements are the same for all of them.

However, it is evident that there is a difference in the physical principles adopted for the functioning, and consequently the function structure of the two products is quite different. In fact, one of them needs pressure and a relative rotation of the fruit, while the second uses an amplified pressure.

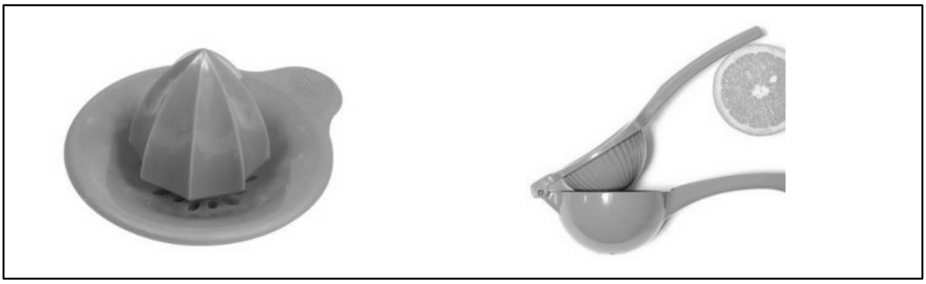


Figure 23. Two types of orange juicer

Supposing to develop for the first time the concept of an orange juicer, one of the two solutions of Figure 23 may be generated and the related functional model can be realized. Then the heuristics of the FSH can be applied and a modular architecture can be developed in the conceptual design phase. Maybe the result is sufficient to pass the evaluation process, however it is difficult to know if another starting concept could be better or worse than the considered one. A possible way to overcome this problem consists in a trial and error approach where multiple product concepts are processed with FSH, and then evaluated. However the designer can't foresee how modularity will affects its proposal during the development of the product concept. Moreover, no support is given to the designer for modifying the starting concept in order to reach a better architecture. Then FSH method can of course be used in conceptual design applications, but the above mentioned peculiarities can be identified in case of original design processes.

It is possible to assert that such a method suffers the same deficiencies characterizing the Pahl and Beitz conceptual design method, i.e. the impossibility to concurrently evaluate different possible concept variants, and then different possible architectures.

Eventually, it is also possible to observe that the method lacks in supporting the definition of the type of modularity.

4.3. Modular Function Deployment

As for FSH, the MFD is based on a comprehensive function modelling of the system which has to be modularized. As explicitly reported in the step two of the method, in order to operate the functional decomposition, it is fundamental to know the technical solutions adopted for the function implementation. Then, according to Blackenfelt (2001) the method is best suited for a redesign of an available product and thus there is a need of more support in the conceptual phase. The reasons which led to this evaluation are reported in the following by considering one of the illustrative case studies present in the book of Ericsson and Erixon (1999).

The considered case study is the application of the MFD to a redesign of a vacuum cleaner, where a set of customer requirements was identified:

- High suction performance
- Low price
- Easy to use
- Long working range
- Low
- Noise
- Easy maintenance
- Easy storage

However, the demonstration of what asserted above can be observed in Step 2 of the approach, where it has been highlighted that the second level of the function structure is a consequence of the technical solutions chosen to implement function belonging to the first level (see Fig. 24).

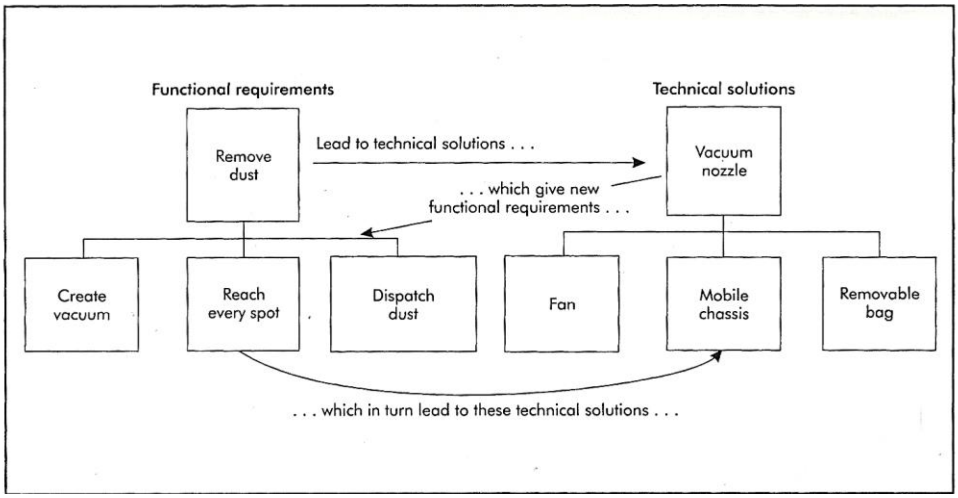


Figure 24. Relationship between the function structure and the technical solutions concerning the vacuum cleaner case study (from Ericsson and Erixon 1999),

Indeed, in that case the first level function was "remove dust" and the adopted technical solution was the vacuum nozzle. Then, the second level of the function structure was created with functions strictly related to the vacuum nozzle. The same can be repeated for the subsequent levels of details. However the method does not provide any specific tool for the development and the selection of the technical solutions. The same Ericsson and Erixon (1999) admit that in order to describe the vacuum cleaner function structure, they used some principles from the axiomatic design.

What stated above implies that the method itself is not sufficient to be used in the concept development of original design processes, but it needs to be assisted by other design tools or methods. Moreover, as for the FSH, no correlation can be identified between modularity and technical solutions until the concept has been developed. Indeed the method only suggests how the proposed technical solution can be grouped into modules, but the development of technical solutions and the modularization are separated procedures.

Furthermore, going on with the analysis of the method, in the step four an instrument is presented to evaluate modules interfaces, but, according to Hölltä-Otto (2005), it is not detailed enough, no useful indication are provided to identify one of the interface types listed by Salvador et al. 2002.

4.4. Final considerations

All the considered methods, although in different ways, can be used during the conceptual design of new products but an iterative process has somehow to be followed. For DSM methods the physical structure of the product has to be determined before the beginning of the modularization process. For those concerning FSH, the functional model can be built during the concept development. However, in order to evaluate different concept variants, the function modeling and the application of heuristics must be performed for each of them. The MFD has been developed to support the definition of modules during the conceptual design, but the method only suggests how the proposed technical solutions can be grouped into modules. Moreover, there is no correlation with modularity during the definition of the technical solutions. Eventually, because of their nature, each of the analyzed methods does not provide any tool for the identification of solutions.

So it can be observed that modularization methods cannot directly take into account the issues related to the choice of the working principles (or the behavior) of the product, because they are made to rearrange functions and structures, already known, according to updated requirements. This kind of general approach is surely useful and the considered methods have been successfully tested in that sense. However, it is possible to argue that in this way the outcomes are strongly influenced by the solutions which constitute the starting product concept. In other words, since the concept development phase doesn’t take into account modularity aspects in a comprehensive way, it cannot guarantee the achievement of the best architecture needed to fulfil the requirements.

Then, it is confirmed that in one hand there are well acknowledged conceptual design approaches with poor capabilities in managing product architecture, and in the other hand there are powerful modularization methods which cannot consider some fundamental aspects of concept generation.

References

Blackenfelt M. (2001) Managing complexity by product modularization. Doctoral Thesis. Department of Machine Design, Royal Institute of Technology, Stockolm, Sweden.

Daniilidis C., Enßlin V., Eben K., Lindemann U. (2011) A Classification Framework for Product Modularization Methods. International Conference on Engineering Design, ICED11

Ericsson, A., Erixon, G., " Controlling Design Variants. Modular Product Platform", Society of Manufacturing Engineers, Michigan, 1999.

Hölttä-Otto, K., "Modular Product Platform Design", Doctoral Dissertation, Helsinki University of Technology. , 2005.

Kusiak A., Huang C.C. (1998) Modularity in design of products and systems. IEEE Transactions on Systems, Man, and Cybernetics - Part A Systems and Humans, vol. 28, no. 1, pp. 66-77.

Pahl, G., Beitz, W.," Engineering Design", 3rd ed., Springer-Verlag, London, 2007.

Salvador F., Forza C., and Rungtusanatham M., "Modularity, product variety, production volume, and component sourcing: theorizing beyond generic prescriptions," Journal of Operation Management., vol. 20, no. 5, 2002, pp. 549–575

Stone, R.B., Wood, K.L., Crawford, R.H.,"A heuristic method for identifying modules for product architectures", Design Studies, Vol. 21, Issue 1, 2000, Pages 5-31.

VDI, 1987 VDI Design Handbook 2221: Systematic Approach to the Design of Technical Systems and Products (translation of 1963 German edition), Verein Deutscher Ingenieure Verlag, Düsseldorf.

5. Guidelines for integration

The performed literature analysis, presented in the previous sections, confirmed that currently it is not possible to systematically face modularity issues before achieving, at least, a partial model of the concept. In other words, it has been confirmed that in case of a specific request concerning the design of a “new” and “modular” product, a comprehensive support is currently missing for early conceptual design activities.

Nevertheless, even if considering the incremental design of existent products (like upgrades or optimizations), there is often the need of introducing solutions implementing new physical principles, new functions and consequently new structures. But even when conceiving these new “partial” solutions, issues related to product architecture may arise, e.g. interaction with the rest of the system and the specific interfaces to be adopted. Also in this case, in one hand current modularity methods can give support only after a preliminary conception of the solutions, and cannot support the designer in conceiving them. On the other hand, in the previous sections it has been shown that functional decomposition and morphology approaches for conceptual design do not provide a sufficient support in facing modularity issues.

Especially in Section 4 it has been shown that despite some similarities, the considered conceptual design model and the three major “families” of modularity methods cannot be matched together in order to overcome the observed limitations. A particular case is that of FSH, which although it is grounded on the EMS functional model and then naturally integrable in the Pahl and Beitz approach, it can be applied only “after” the EMS modeling of the concept.

But at this point, a question certainly arises to the reader: “Why other existent design approaches have not been considered beyond functional decomposition and morphology?”

The analysis has been restricted to the above mentioned approaches not only for their extremely wide diffusion in academia, but also because, despite the observed lacks, provide a comprehensive support to the designer during most of the phases constituting conceptual design. Differently, other design models and/or methods, even if characterized by some not negligible advantages, do not cover the entire process concerning the conception and the formalization of concepts. For instance, considering Axiomatic Design (Suh 1998), it is possible to assert that is characterized by an high diffusion in practice, however, according to Shai et al. (2009) it lacks in assisting the designer during the synthesis of solutions. Then it cannot been applied in early conceptual design activities, indeed the design matrix (which connect design parameters to functional requirements) is still unknown. But other novel approaches for design have been identified, e.g. the SOS of Ziv-Av and Reich (2005), the Infused Design of Shai et al. (2009) or the Parameter Analysis of Kroll (2013). However, the SOS can be efficiently used if concept's building blocks are already known, but cannot support in conceiving them. Indeed, the authors merely quoted TRIZ (Altshuller 1984) for conceiving novel building blocks. Infused Design is a powerful approach capable to get knowledge from different disciplines and to translate it into creative concepts. However it is focused on the communication between experts belonging to different disciplines, not in conceiving the single ideas or building blocks of the overall solution. Eventually, the Parameter Analysis is claimed by Kroll (2013) as a an approach capable of overcoming the lacks observed by literature in functional decomposition and morphology approaches (see Section 2). However it seems very hard to systematically face modularity issues during the concept development process characterizing such a proposal.

Then, it can finally be inferred that the current contributions concerning conceptual design cannot be used for a comprehensive merging with existent modularization approaches, aimed at considering modularity since earliest creative activities of the design process. Nevertheless, trying to modify the modularization approaches seems to be useless, since they have been though only for "reorganization" of something already existent.

This is the reason that led the author of this thesis toward the development of a new methodological proposal, capable of overcoming the lacks observed in Section 2 for functional decomposition and morphology, and allowing a systematic management of modularity since early concept activities.

Here in the following a detailed description of what is intended to be developed is reported.

5.1. Ideal characteristics of the new proposal

On the base of what asserted before, an hypothetical new methodological proposal, aimed at managing modularity and at overcoming the lacks observed in the reference conceptual design model, should be characterized by the following points:

- Identify early in the conceptual design process, the needs to apply modularity.
- Guide the designer in generating correct modular solutions "before" knowing the overall solution, neither in terms of functions.
- Avoid to move toward a maximization of modularity, but allow to reach the optimal product architecture, in terms of satisfaction of the requirement list characterizing the specific design task.
- Reduce/avoid prejudices in decomposing a design problem.
- Allow to easily generate and evaluate different possible concept variants characterized by different functionalities and different working principles.
- Allow keeping track of the information gathering activities involved in the decisions taken during the conceptual design process.
- Visualize the design space exploration.

The above listed characteristics constitute the “requirement list” which guided the author of this thesis in the subsequent research activities, aimed at developing the new methodological proposal. However such a goal requires a noticeable amount of resources in order to be fully achieved. Indeed, once a first version of theoretical base of the proposal has been developed, a series of tests has to be performed, both in academia and in industry, in order to obtain the necessary information for further developments. But due to the limits of resources characterizing the present activity, the actual goal faced in this thesis has to be reconsidered. In the following sub-section, a detailed description of what is the actual goal expected for the scope of this thesis is reported.

5.2. Expected results for the scope of the present work.

The principal limits characterizing the research activity performed in the present PhD scholarship are the limited amounts of time and human resources. Indeed, it is very hard to obtain a comprehensive new method for conceptual design and product architecture in only three years and with human resources limited to the sole Phd student. Moreover, the constant necessity of evaluating the intermediary proposals can be fulfilled only with well-structured tests and/or realistic applications. But in the present work, due to the lack of a direct involvement of industry, the above mentioned tests and applications have been performed by considering samples of convenience restricted to engineering students.

So, taking into account the above mentioned limitations, it is possible to infer that a comprehensive validation of the new proposal is not expected here. However, the evaluations that will be performed are expected to give first assessments about the performances of the new approach. More specifically, specific tests will be performed in order to investigate on the following two points:

- Potentialities of the new approach in generating modular concepts.
- Potentialities of the new approach in supporting the designer in the generation of new concepts.

For what concerns the fulfillement of the ideal requirements listed in the previous sub-section, they constitute a sort of guideline to be followed during the development of the proposal. Discussions about the actual matching between expectations and obtained results will be perfomed on the base of available material.

References

Altshuller, G. S. 1984. Creativity as an exact science: The Theory of the Solution of Inventive Problems. Gordon and Breach Science Publishers.

Kroll, E. (2013). Design theory and conceptual design : contrasting functional decomposition and morphology with parameter analysis. Research in Engineering Design, 165–183. doi:10.1007/s00163-012-0149-6

Shai, O., Reich, Y., & Rubin, D. (2009). Creative conceptual design : Extending the scope by infused design. Computer-Aided Design, 41(3), 117–135. doi:10.1016/j.cad.2007.11.004

Suh, N. P. (1998). Axiomatic Design Theory for Systems. Research in Engineering Design, 10, 189–209.

Ziv-av, A., Reich, Y. (2005). SOS e subjective objective system for generating optimal product concepts. Design Studies, 26, 509–533. doi:10.1016/j.destud.2004.12.001

6. Development of the proposal

As highlighted in Section 4, the results obtained by the application of the modularization methods are strongly influenced by the solutions adopted in the starting concept, which however has been developed without considering modularity. In fact, it has been inferred that the available modularization methods can assist concept design activities only with a trial and error approach, where a product concept has to be formerly developed. Nevertheless, Fixson (2003) observed that the need for a modular architecture may arise during the definition of the requirement list as well as early in the design process. Furthermore, Ulrich, Eppinger (2003) state that modularity issues may arise even during the concept design phase, although only informally. Such a statement required further investigations aimed at verifying when and why modularity issues emerge during conceptual design. Indeed, the outcomes of such an activity could contribute to the understanding of how modularity issues can be managed during the conceptual design of a new product. The proposed investigation method is based on a problem-solution analysis, based on the following two definitions:

- Modular problems: this parameter refers to those design problems in which their resolution could take advantage from one or more modularity benefits acknowledged by literature (Table 1).
- Modular solutions: they represent solutions whose characteristics can be attributed to well acknowledge types of modularity (Table 2). Thus, every investigated solution which presents one or more characteristics belonging to a standard group of modularity is assumed to be a modular solution. It is worth to notice that a modular solution may also be something not completely identifiable in a module.

Modular problems are used for the identification of design problems potentially solvable with modularity, while the modular solution definition is employed to discern modularity in the technical solutions adopted in the considered product. A comprehensive description about the above mentioned investigation approach and the obtained results is reported in Section 7. However, for the scope of this section it is sufficient to report that by using modularity benefits as tools for the identification of modularity, an effective relationship has been observed between the identified modular problems and the related modular solutions.

On the base of these preliminary observations, the development process of the proposal has been initiated. The description of the theoretical part of such an activity is reported here in the following, while the performed test are deeply described in Section 7.

6.1. A preliminary proposal for non-structured design activities.

Considering the observations expressed in Section 4, it has been started the development of a new design approach capable of taking into account modularity issues since early concept generation. However, due to the impossibility to find something compatible with current conceptual design approaches, a first version thought for non-structured conceptual design activities has been previously tested. Such a proposal can be summarized into two main parts:

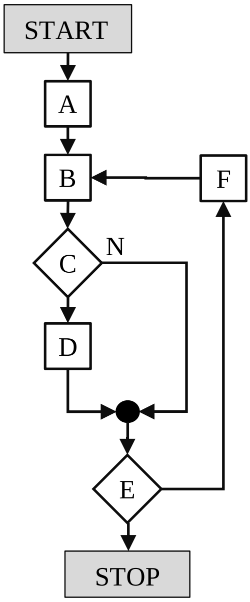
- Identification of the modularity needs before starting the (non-structured) concept generation process.
- Support in generating modular solutions for each identified modularity need.

Here in the following, a detailed description of the two parts is reported.

6.1.1.Identification of modularity needs

The first part of the proposal takes inspiration by the benefits attributed to modularity, which have been listed and described in Section 3. More in particular, these benefits are considered here as potential instruments to be used in order to ease the fulfilment of specific design requirements. Such a idea was born by observing the results of the investigation analysis performed around the rise of modularity in conceptual design processes (see Sub-section 7.1 for details). For the scopes of this section it is sufficient to premise that in such an investigation it has been observed a sort of relationship between modularity benefits and the presence of modular solutions. Then, such a peculiarity has been used to create the above mentioned identification tool. For instance, if considering a requirement concerning the lowering of the product manufacturing costs, it is quite simple to deduce that the "Economy of Scale" or the "Logistic optimization for production/assembly" benefits, as described in Section 3, can give a potential help. Since these benefits are associated to modularity (see Table 1), it means that in order to fulfil the specific requirement, the use of modular solutions has to be evaluated, at least.

Then, the identification process has been formulated and a representative schematization is reported in Figure 25, and it is intended to be repeated for all the requirements of the design task. Then, once a specific requirement has been selected, it has to be compared with the list of modularity benefits (Table 1) in order to find compatibilities. If no matching can be observed, the requirement is considered as “non-modular” and then it can be faced normally. Otherwise, if even a single compatibility can be observed, the requirement is considered as “modular” and a specific procedure has to be pursued in order to find related solutions. More specifically, a modularity benefit and a requirement are considered compatible when in the definition of the benefit it is possible to find a chance to aid the fulfilment of the requirement. Then it is possible to observe that the identification of the opportunities to use modularity in concept generation is operated only by considering the requirement list, thus this task can be performed before to start generating concept ideas.



A	Select a product requirement
B	Compare the requirement with the modularity benefits
C	Is the requirement compatible with any modularity benefit?
D	Proceed evaluating the possibility to generate a modular solution
E	Are other requirements to be examined still present?
F	Select one of the remaining requirements

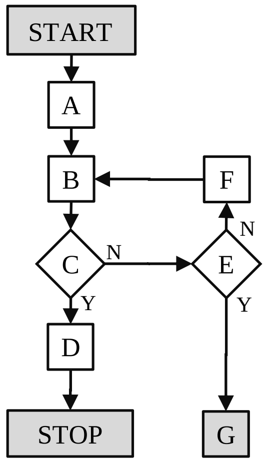
Figure 25. Identification process for modularity opportunities.

6.1.2.Generating modular solutions

Once an opportunity to use modularity has been identified with the logic introduced above, the designer should conceive befitting modular solutions. The procedure introduced here aims at giving a support for performing this activity, and takes inspiration from the literature contributions concerning the classification of the modularity types reported in Section 3 (Table 2):

- Interfaces types of the modules. Describing the characteristics of the connectivity among the components of the system.
- Interactions within the system. Describing how the modules are matched together in order to form the system.
- Supply type of modules. Describing the way by which the components of the systems are provided.

Then, standard modularity types are used as a sort of design catalogue in order to inspire the designer during the development of the specific solution. More precisely, for each design requirement in relationship with one or more modularity benefits, the designer is asked to follow the roadmap of activities described in Figure 26.



A	Select a group of modularity type.
B	Consider one of the possible type variants belonging to the same grouping criterion.
C	Does the modularity type variant inspires any idea?
D	Conceive solution details concerning the other two modularity groups by considering related standard type variants.
E	All the three modularity groups have been already considered for generating the solution?
F	Select one of the remaining modularity groups.
G	Proceed without considering modularity for this requirement.

Figure 26. Logic of the experimental modular solution generation process

Referring to that picture, the designer has to think about some “ideas” capable to fulfil the identified modular requirement, and for this purpose, he/she has to select one of the standard modularity groups of Table 2, and think about a possible implementation. Once the idea has been focused, the solution has to be completed by defining the other two characteristics. Indeed, the interfaces must be defined, as well as the interactions between the various components and the way the hypothetical modules have to be supplied.

In case of impossibility to conceive modular ideas with the selected category, the designer has to try with other ones, but if even in this way it is possible to find modular solutions, the designer can proceed without considering modularity for the specific requirement.

6.1.3. Considerations about this first version of the proposal

The aim of the proposal described above, was to furnish a first evidence about the possibility to face modularity issues before any concept generation process. A preliminary test, described in Section 7, effectively shows not negligible effects in supporting modular solutions, and moreover, gives a preliminary validation on what concerns the identification of modularity opportunities. However, accordingly to the available resources, the test was performed in a simplified manner. Indeed, in the design task it was explicitly specified which the requirements to be faced with modularity were.

But the absence of a structured design approach certainly leads to difficulties in managing, for example, multiple modularity needs. Indeed, it can be very hard to combine different solutions, modular or non-modular, conceived for specific requirements. However, none of the acknowledged conceptual design processes seems to be compatible with the proposal. Then it has been evaluated the possibility of developing a new conceptual design method, capable to implement the fundamental logic described above for achieving modularity.

6.2. A new problem-oriented conceptual design approach.

In order to find a possible way to implement the proposal for modularity and to overcome the lacks of functional decomposition and morphology approaches reported in Section 2, the development of a new approach for conceptual design has been started. The leading requirement of such an activity was to implement the modularity approach formulated in the previous sub-section, and since it was based on a definition of modular problem and solutions, attention was focused on a problem-solution model of the design process. However it is worth of notice that it does not imply that the design process is considered here only as a problem solving activity. Indeed recent studies reports that the design process cannot be assimilated to a mere problem solving activity, because the first is characterized also by other important features. i.e. the unexpected expansion of the initial concept, the fundamental role of social interactions and the use of learning devices (or processes) to gather information (Hatchuel 2002). Nevertheless, it is possible to assert that problem solving is a fundamental part of the design process, and despite the presence of theoretical issues which have still to be solved, it is currently accepted as the normal language for talking and thinking about design (Dorst 2006). As a matter of the fact, Cross (2000) states that designer difficulties is two-fold i.e., understanding the problem and finding a solution. Therefore, this work is mainly focused on the development of a problem-solution approach for conceptual design, but the author is conscious of other important aspects actually characterize the design process.

It is possible to formulate design problems concerning how to implement functions, how to ensure a certain behavior or how to reach desired performances. Therefore, functions, working principles and geometrical solutions can be generated with the aim of solving specific design problems at different levels of abstraction. Thus, differently from a pure functional decomposition approach, considering the design problems, whatever they concerns functions, physical principles or simple geometrical solutions, it is possible to visualize the relationship between solutions and specific problems which they solve. Keeping track of the link between problems and related solutions is considered here as a valid help in reducing the occurrences of preconceptions or prejudices in problem decomposition.

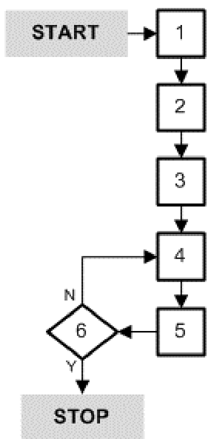
More precisely a design problem is considered here as any "question" expressed in the form "How to *verb - noun?*" (e.g. "How to generate power?" or "How to transmit torque?", or even "How to ensure adaptability?").

The set of information for formulating the problems characterizing the design tasks are extracted from the product requirement list, which is intended here as composed by design objectives and constraints, both subdivided into functional and non-functional ones. In other words, from the requirement list the designer has to identify the functionalities of the system (desired and/or compelled), the performances of the system (desired and/or compelled), the quality of the product (desired and/or compelled), the production characteristics (desired and/or compelled), and to translate them into design problems to be solved. However, in order to obtain a structured design process, this task cannot be accomplished without considering a logic equipped by a predetermined set of rules.

On the base of what expressed above, the model developed in this work aims at proposing a tool for generating and considering many possible solutions at different levels of abstraction, keeping track of problem formulation and decomposition, together with the information gathered during the conceptual design activities. Such a model considers the conceptual design process as subdivided into three sub-phases, namely the *concept generation*, the *concept composition* and the *concept selection*. The outcomes of the concept generation phase are the single (partial) solutions related to each problem obtained from the decomposition of the overall one. In concept composition, different combinations of (partial) solutions are considered in order to obtain the concept variants, represented by sketches or rough cad models. Finally, in concept selection, a set of preferred concepts is selected by means of various evaluation parameters. However, the focus of this work is primarily on the concept generation and secondly on the concept combination. While, concerning the concept selection nothing new is proposed here and well acknowledged literature techniques can be considered fully applicable (e.g. those mentioned in Section 2).

6.2.1.Proposed concept generation approach

Before starting with the description of each part composing the proposed concept generation approach, it is worth to introduce its overall logic in order to provide a general overview to ease the understandability of the contents. As shown in Figure 27, the method can be summarized with five main general activities and a verification node.



STEP	Description
1	Main task identification and formulation
2	Main problems formulation
3	Main problem analysis and resolution
4	Formulation of solution dependent sub-problems
5	Sub-problems analysis and resolution
6	Has the main task been accomplished?

Figure 27. The overall logic of the proposed concept generation approach

First of all the main design task has to be identified and formulated (Step 1), in order to orient the activity towards the right direction. Subsequently, one or more design problems, not dependent each other, are formulated (Step 2) and tackled in order to find a first level of solutions (Step 3). Once one or more solutions have been generated for the main problems, other sub-problems can be formulated for each identified solutions (Step 4), e.g. concerning how to implement them. Then in Step 5, a second level of solutions is generated for such a second level of problems. But, depending on the abstraction level of both sub-problems formulated at Step 4 and related solutions generated at Step 5, there is the possibility to find other sub-problems belonging to a lower level of abstraction. This event implies that the process moves to the loop represented in Figure 27 by steps 4 and 5, which is controlled by the verification step6.

However, all the necessary rules and the particular activities involved are not shown in such a representation, which is only a very simplified schematization of the proposed conceptual design process, aimed at giving a rapid and simple interpretation of the overall logic. Detailed descriptions of each part developed in this work are reported in the following.

6.2.2.How to face single problems: the ASE process

The core of the proposed concept generation approach, i.e. a logic concerning problem analysis and solution (Steps 3 and 5 of Figure 27), takes inspiration from Watts (1966). In the Watts model (as quoted in Evbuomwan et al. 1996), three processes of "analysis", "synthesis" and "evaluation" are performed cyclically from a more abstract to a more concrete level of detail of the developed solution. It is worth to notice that the above mentioned processes are the same introduced by Jones (1962) (as stated in Evbuomwan et al. 1996), while the definitions considered here, thought for facing single design problems, are the following ones:

- Analysis (A): the design problem is analysed in order to evaluate if the owned information are sufficient to completely understand it. Then, the possibility to decompose it into solution independent sub-functions is investigated.
- Synthesis (S): the available information is examined in order to evaluate the possibility to solve the problem. Subsequently, if possible, potential solutions are generated for the considered problem.
- Evaluation (E): available information is examined in order to verify the presence of an exhaustive set of evaluation parameters to be used in order to accept or reject the proposed solutions. If information is sufficient, a verdict is emitted for the examined solution.

Moreover, the model developed here, beyond the possibility to generate a variety of solutions for a single problem, aims at mapping the information gathering during the A-S-E activities (see Figure 28 and Table 5).

Then, referring to Figure 28 and Table 5, a short description of the steps composing the ASE process is here reported. Firstly, a design problem must be selected from the set of those which have to be faced during the activity (Step A). Then it is possible to observe that four groups of actions and verifications exist, namely the Analysis, the Synthesis, the Evaluation and the Information Gathering ones. Here in the following, the three main groups of the ASE are described in detail by means of dedicated paragraphs, each of them comprising a description of the involved Information Gathering activities.

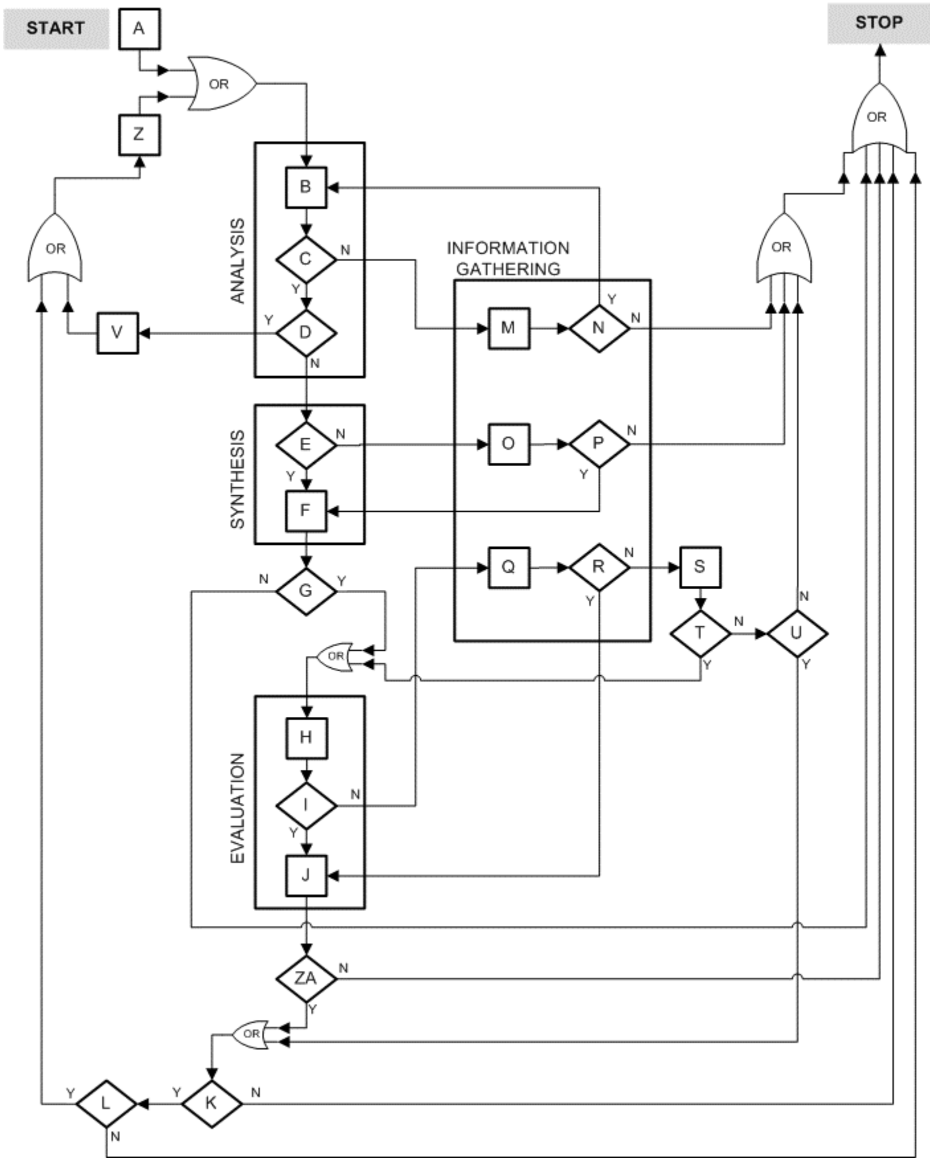


Figure 28. ASE logic of the activities involved in a solving process of a single problem.

Table 5. Short description of the steps shown in Figure 28.

A	Select a design problem
B	Analyse the problem
C	Is it possible to completely understand the problem?
D	Is it possible to decompose the problem into solution independent sub-problems?
E	Is the know-how sufficient to try to solve the problem?
F	Try to propose a solution
G	It has been possible to find solutions?
H	Select a solution to be evaluated
I	Are information sufficient to evaluate the selected solution?
J	Accept or discard the evaluated solution
K	Has the problem selected in A been decomposed?
L	Do other sub-problems to be processed still exists?
M	Acquire the required information in order to understand the problem
N	Are the upgraded information sufficient to understand the problem?
O	Acquire the know-how for the problem solving activity
P	Is the upgraded know-how sufficient to solve the problem?
Q	Acquire evaluation parameters for the considered solution
R	Are the upgraded evaluation parameter sufficient to evaluate at least one solution?
S	Discard the evaluated solution
T	Are other solutions to be evaluated still present?
U	Does (at least) a solution exists for the considered problem?
V	Formulate the sub-problems
Z	Select one of the formulated sub-problems
ZA	Does (at least) one solution has been accepted?

Analysis group

Then, entering in the Analysis group, the Step B consists in analysing the problem, i.e. trying to understand which is the need to be fulfilled. In order to understand the problem, there is the need of a certain type of knowledge, then the verification at Step C appears, with the aim of evaluating if an information gathering activity is needed. If more information is needed, the process jumps to the Information Gathering group, more in particular to Step M, followed by the verification in Step N. Obviously, if it results impossible to acquire the necessary information, the ASE process ends, since it is meaningless to proceed further. Differently, if the upgraded information is sufficient, it is possible to evaluate the possibility of a solution-independent problem decomposition (Step D), i.e. a subdivision into more elementary problems which are not derived from any prejudice concerning the final solution. Such a particular problem decomposition implies that in order to solve the main problem, each generated sub-problem has to be solved. This is quite obvious since leaving unsolved even a single sub-problem means that only a part of the main problem has been solved, condition which is not acceptable in engineering design processes.

If there is the possibility of such a decomposition, the next step is then to formulate the sub-problems (Step V) and then to select one of them (Step Z) in order to proceed again with the ASE process. Differently, if no solution-independent decomposition is possible, the process moves directly to the Synthesis group.

Synthesis group

The first step of the Synthesis group is Step E, where the designer has to evaluate if the available know-how is sufficient for conceiving one or more solutions. In case of negative answer, the process moves again into the Information Gathering group, precisely on Step O and subsequently on the verification of Step P. Even in this case, if it is not possible to acquire the necessary know how for generating any type of solution, the process stops even if other sub-problems to be processed exist. The reason of such a drastic decision lies in two assumptions. Firstly, if the main problem is decomposed as described above, each solution independent sub-problem has to be solved in order to solve the main one. Secondly, the scheme of Figure 28 represents an ideal activity where everything possible has been done both for acquiring information and for solving problems. Then, if it is not possible to solve a sub-problem, on the base of the first assumption it is considered meaningless to proceed further. But if the information is considered sufficient the process goes to Step F, where the designer has to try to generate solutions. Then, if it is possible to find one or more solutions the process continues, conversely it stops even if other sub-problems to be processed exist.

Evaluation group

At this point, it is the time to enter in the Evaluation group in order to assess the generated solutions. In Step H one of them is selected from the obtained set. In this case, information concerning the evaluation parameters is needed, and if the available one is considered not sufficient, the process moves again towards the Information Gathering group, more precisely on Step Q. Then, if the upgraded information is considered sufficient in order to make a decision, the solution is accepted or rejected (Step J). Differently, if it is not possible to make a decision, the solution has to be discarded (Step S). Indeed, the author of this thesis believes that in the context of engineering design processes, if it is impossible to evaluate a solution with a sufficient level of accuracy, it has to be discarded.

After Step S, thanks to the verification at Step T, the evaluation can be performed on all the solutions composing the generated set. But when all the set has been examined, in Step U it is verified that at least one solution has been conceived. Indeed, if it has not been possible to solve the problem (or the sub-problem), the process stops (see first and second assumptions described above about the process schematization). After accepting or rejecting all the solutions, if no one can be accepted, the process stops even if other sub-problems have to be faced. On the contrary, if at least one solution has been accepted for the problem, in steps K and L it is verified if other solution-independent sub-problems have to be processed.

Concluding remarks on the ASE logic

Summarizing, the proposed ASE model takes into consideration the possibility of needing more information to understand the problem and the eventuality of a further decomposition in more elementary (solution independent) sub-problems. Beyond that, it is considered also the possibility of the need to acquire additional know-how to solve problems and defining new specific requirements to be used as further evaluation parameters for a conceived solution. The scheme shown in Figure 28 refers to a generic problem solving process and aims at representing the sequence of activities and questions to be faced for a "single problem". Such a scheme, despite its apparent complexity, aims at describing an intuitive problem solving process, i.e. understand the problem, synthetize potential solutions and evaluate them.

But as highlighted above, the ASE logic has been thought for facing single problems, not considering the relationship between generated solutions and other problems which they also generate. However, as expressed in Section 2, keeping track of the relationships between problems and solutions can be helpful when facing design tasks, allowing a better management of the variety of proposed solutions and giving the possibility to evaluate how the concept has been obtained. Therefore, to make possible a practical application of the ASE logic in a real context, it is necessary to keep track of the problem-solution relationships concurrently with the information gathering process. In order to do that, some fundamental tools and rules are needed. Here in the following, a detailed description of them is reported, which constitutes the other fundamental part of the proposed approach.

6.2.3. The Problem-Solution Network as reference model for mapping the design process

The use of graphical tools or diagrams in engineering design is widely diffused, thanks to their capability in facilitating human cognitive processes (Aurisicchio and Bracewell 2013). Such a not negligible characteristic has been taken into consideration by the author when facing the need of mapping the conceptual design process and managing the information gathering activities. Indeed, the author encountered the necessity of an elementary graphical tool, capable of visualizing the problem-solution coevolution, and also capable of keeping track of the information gathering activities.

Hierarchical tool for mapping the design processes already exists in literature, e.g. Function-Means tree of Andreasen (1980) (as quoted in Aurisicchio and Bracewell 2013), or the decision tree of Marples (1961). However, even if an interesting extension was developed in order to include design history information (Malmqvist 1997), the Function-Means tree was developed for functions, avoiding the possibility of considering generic design problems concerning for example feasibility, performances or quality requirements. The Marples decision tree is well suited for representing generic problem-solution relationships, however it was not conceived to keep track of information gathering. But another tool with similar characteristics exists in the OTSM-TRIZ (Khomenko et al. 2007) base of knowledge (OTSM is the Russian acronym for the General Theory of Powerful Thinking), i.e. one of the modern directions of development of the well-known TRIZ method (Altshuller 1984). Such a tool is the Network of Problems (NoP) (Fiorineschi et al. 2011), and is constituted by a hierarchical graph, where problems and solutions represented by boxes are linked together by means of arrows. The NoP is part of a more extended process called Problem Flow Network (PFN), but is not in the scopes of this work to give a thorough description of such a problem solving approach. However, the fundamental logic of the NoP, i.e. the decomposition of a problem into more elementary sub-problems and the connections of them with the related solutions, has been taken as a reference in the development of the main tool of the proposed model. A schematic representation of such a graphical tool, hereinafter called “Problem-Solution Network” (PSN), is reported in Figure 29 and described here in the following.

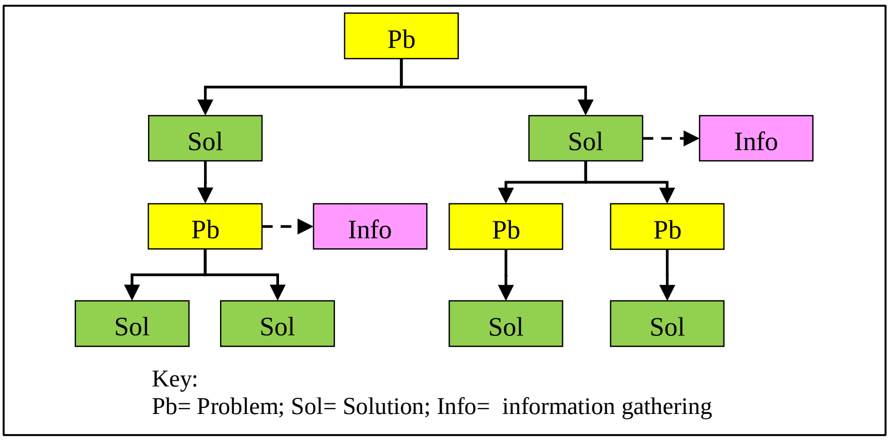


Figure 29. Generic example of the PSN.

Similarly to the network proposed by Marples (1961) (see also Evbuomwan et al. 1996), by means of the PSN, it is possible to keep track of how problem decomposition is influenced by solutions. Indeed, taking as a reference the generic example of Figure 29, the overall problem (i.e. the first from the up of the network) is connected with related potential and distinct solutions. Each of them leads again to different problems, which in turn are solved with other distinct solutions belonging to a lower level of detail and/or abstraction, and so forth. However, differently from the Marples tool, the PSN allows the use of the "info" boxes, which constitute a simple way to manage the information gathering activities during the whole conceptual design process. They can be connected either to problem boxes or to solution boxes, since, as previously described, in both cases there is the possibility of a need of information (see Figure 29).

By means of the graphical tool described above, the designer can keep track of the information gathering and its relationship with problems and solutions. Furthermore, he/she can visualize the branches of the network leading to a major need of information. In this way, even before composing the complete concept, it is possible to estimate the development efforts related to specific problem-solution branches and comparing them with constraints like time and costs, so as to support decision making activities related to the subsequent design steps.

6.2.4. Rules for composing the PSN

The ASE process applied recursively to each problem represented on the PSN can lead to an uncontrolled expansion of the network, both in vertical (solutions even more detailed) and in horizontal directions (exploration of the design space even more deepened). Such an "inconvenient" must be avoided, because it may lead to the impossibility of managing the design process. This is the reason why a set of rules has been conceived, to control the network expansion, i.e. the level of detail of the solutions and the extent of exploration of the design space. The set of rules are listed below, and a detailed description of them is given in the following:

- Main task formulation.
- First level problems formulation.
- Solution-independent problem decomposition.
- Follow the correct sequence of abstraction levels.
- Independency of PSN branches.
- Completeness of the PSN.

Main task formulation

Such a rule aims at guiding the designer towards the definition of the overall task (the first problem box shown in Figure 29 reading it from the top). In practice, it must be ensured that the PSN is created starting from the highest level of abstraction and considering the right level of detail of the system to be developed. The designer has to think to the desired outcomes of the design activity, i.e. the results he/she wants to obtain through the ideation process. Therefore, in performing the definition of the main task, the attention should be directed towards the main objective that the design activity should satisfy, without prejudices concerning feasibility, performances, technical properties of the system, features of existing systems that perform the same function and modalities to achieve the expected results. The designer should accomplish such a reflection by asking for “What I have to design?” in such a way any bias related to his/her background and skill, which can limit the generation activity, can be avoided. For instance, considering the design task related to the design of a cutting zone for a core cutter machine, independently from the requirement list, the designer has to “frame” the type of activity. In such an example it consists in simply “design a new cutting zone of the system”. It is important to not confuse the task with “design a new core cutter machine” or even “design a new blade”, because beyond the evidence that the task can be extremely different, an incorrect formulation may leads to design prejudices or even to a waste of resources. Indeed, the development of a new blade may be one of the key solutions of the actual problem solving process, but focusing the activity on a single part of the system is self-limiting. Similarly, extending the design task to the whole core cutter machine may lead the designer to lose the focus of the project.

First level problems formulation

Once the design task has been correctly formulated, the process proceeds by defining the principal problems to be faced, i.e. the first level of solution independent problems derived from the main task. The PSN represented in Figure 29 is not suitable for such a task, since it starts by proposing solutions directly for the main problem box. Indeed, in this way may result difficult to keep the right level of abstraction and to avoid bias when proposing solutions.

The rule described here aims at decomposing the main design task into solution-independent problems with the highest level of abstraction. Similarly to the classical functional decomposition approaches, it means to identify “what the system does” by observing it like a black box in which something enters and consequently something “changed” exits. In order to to avoid prejudices, it is extremely important to not consider anything related to “how” such changes happen. For instance, considering again the example of the development of a new cutting zone of a core cutter machine, the first level of the PSN become that represented in Figure 30.

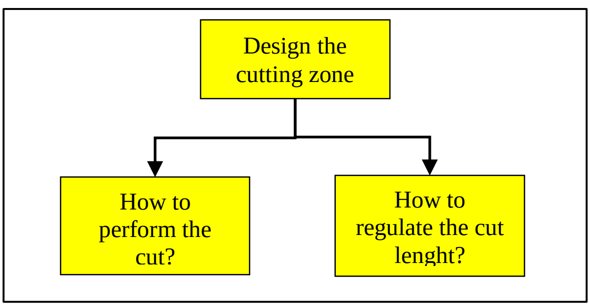


Figure 30. Example for the first level of problems of the PSN. The first box from the top represent the design task, while the other two represent the problems concerning the main functionalities of the system to be designed.

Functional problems expressed in Figure 30 are absolutely solution independent, because in any case, the core has to be cut and the length of the cut piece has to be regulated. Such problems are considered independent each other, and with the requirements list obtained for the considered case, it is not possible to formulate any other solution-independent problem belonging to the same level of detail. Because, for instance, the problem “How to move the core towards the cutter?” is not a solution independent problem, since it implies to assume that the cutting system is fixed, and the core is moving. In this way, it is difficult to imagine solutions where the core is stationary and the cutting system is moving, but some systems based on this principle already exist. Howeve, it does not mean that such a problem (and similar ones) is not considered in the PSN, but only that it will be formulated, maybe even for multiple times, for specified solutions at a lower level of abstraction.

By starting from the above described solution-independent problems, the ASE logic can be applied on each of them, allowing the development of the related problem-solution branches. However, the development of the network requires further important rules described in the following.

Solution independent problem decompositions.

As recalled, one of the key features of the proposed approach is the possibility of keeping track of the relationships between problem decompositions and partial solutions, as shown in Figure 29. However, and especially when facing high levels of abstraction, the ASE logic foresees the possibility of further solution-independent problem decompositions. In such an eventuality, it is sufficient to add a new problem level in the PSN, as shown in Figure 31.

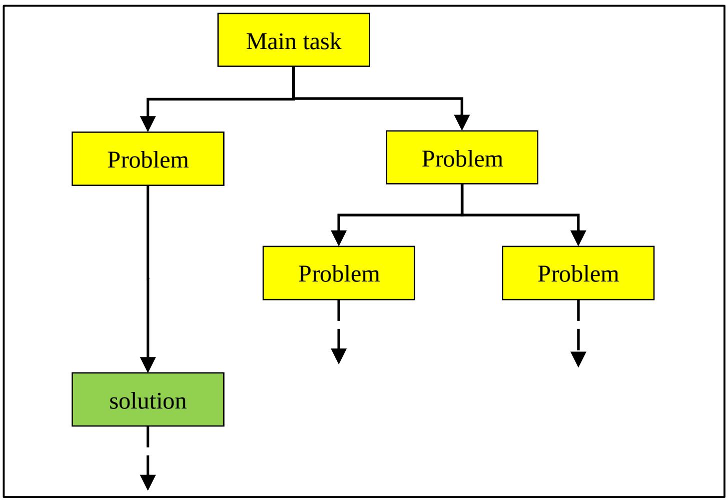


Figure 31. Additional level of problems to be used in case of solution-independent problem decomposition.

Follow the right sequence of abstraction

It is acknowledged that abstraction may play a crucial role in conceptual design (Bonnema and Houten 2006), however it seems that only expert designers have the ability of its correct use. More precisely, abstraction is important to achieve the correct problem decomposition and to explore the design space at the maximum extent, since it allows the designer to avoid focusing only on already known solutions. Indeed, especially to non-experts designers, it often happens to propose a solution with a too low level of abstraction, avoiding to see other possible solutions characterized even by completely different principles. For instance, preliminary tests conducted by the authors with engineering students, shown that they often limit the exploration of the design space to a single solution when facing a specific problem, and try to develop it in detail. Such a behavior can be mapped through the PSN, and the differences on the exploration of solutions between a superficial abstraction and a deeper one can be easily observed (Figure 32).

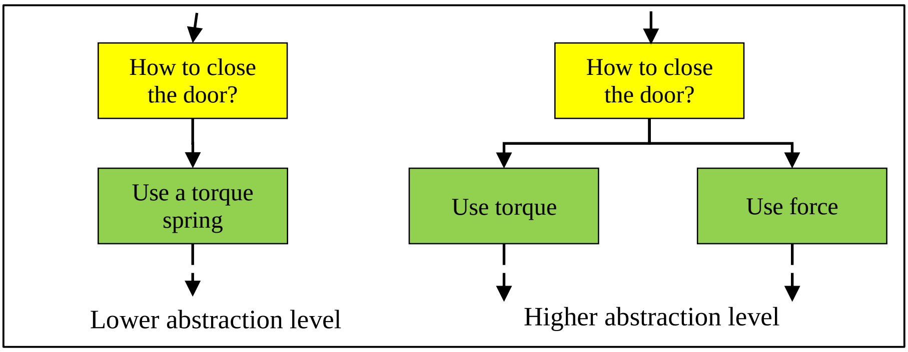


Figure 32. Different level of abstraction used to solve a design problem.

For instance looking to the excerpt of PSN represented in Figure 32 on the left, it is possible to infer that other solutions are completely missed by the designer, e.g. the use of a compression spring and a cam system (commercial solution), or the use of magnets to generate torque. While, from the excerpt of PSN represented in Figure 32 on the right, it is possible to see that any possible solution, not only belonging to the “torque” branch, but also exploiting a force may be generated.

Another possible drawback imputable to a lack of skill in using abstraction may arise when a too low level is considered for formulating the problem. For instance, considering the correct example of Figure 32 (on the right), for the solution “Use torque”, it is not correct to formulate a problem like “How to dampen the spring action?” The reason is that this kind of problem formulation implicitly considers the usage of a spring, while other solutions may exist (e.g. magnets or a rotary motor). Indeed, it is necessary to previously formulate a problem concerning “How to generate torque”, as represented in Figure 33.

So the present rule can be applied through two “questions” that the designer has to ask himself when proposing a solution and when formulating a problem:

1) “Does the formulated problem implicitly consider the adoption of a specific solution at lower level of abstraction?” (e.g. starting from the solution “Use torque”, if formulating the “How to dampen the spring action?” problem, implies the use of a spring, which is a solution at a lower level of abstraction). In case of positive answer, identify such a solution and individuate the actual problem that the solution solves (e.g. the spring may solves the “How to generate torque” problem). Then realize the complete problem-solution sequence in order to arrive at the formulated problem (see for example Figure 33). However there is the possibility of formulating problems which implicitly considers more solutions and related problems at different levels of details. In such a case the above mentioned process must be applied in more steps, as represented schematically in Figure 34, in order to obtain a complete problem-solution sequence.
2) “Does the proposed solution belong to a more extended (or more abstract) family of solutions?” (e.g. the torque spring belongs to the “use torque” family). In case of positive answer, the designer has to consider the whole family of solutions. After that, he/she has to understand which is the problem that the formerly proposed idea is actually solving (e.g. the torque spring solves the “How to generate torque?” problem). Such a problem has to be investigated with the question 1, and in case of presence of implicit solutions, the related “decomposition” process has to be applied in order to obtain the formulation of the problem that the proposed solution actually solves. A generalized schematization of what stated above is shown in Figure 35.

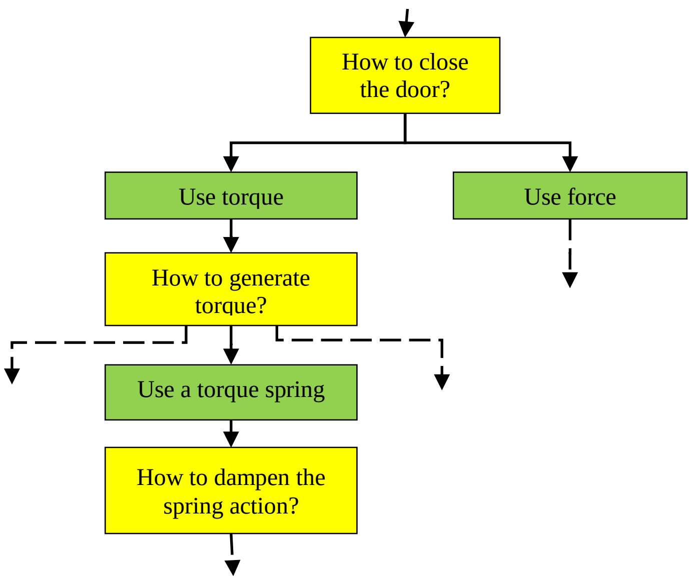


Figure 33. The correct problem-solution sequence for considering a torque spring in order to solve the problem “how to close the door”.

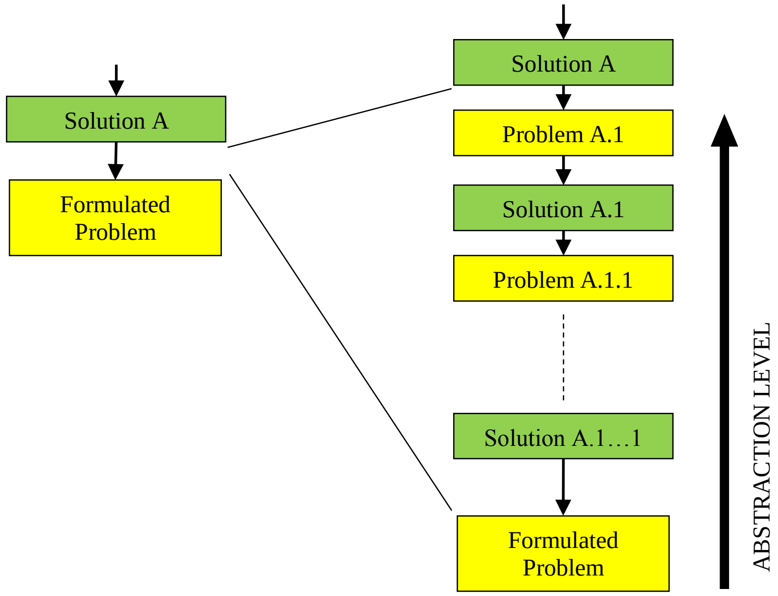


Figure 34. Schematic representation of a generic case where the formulated problem implicitly consider more solutions at different levels of abstraction.

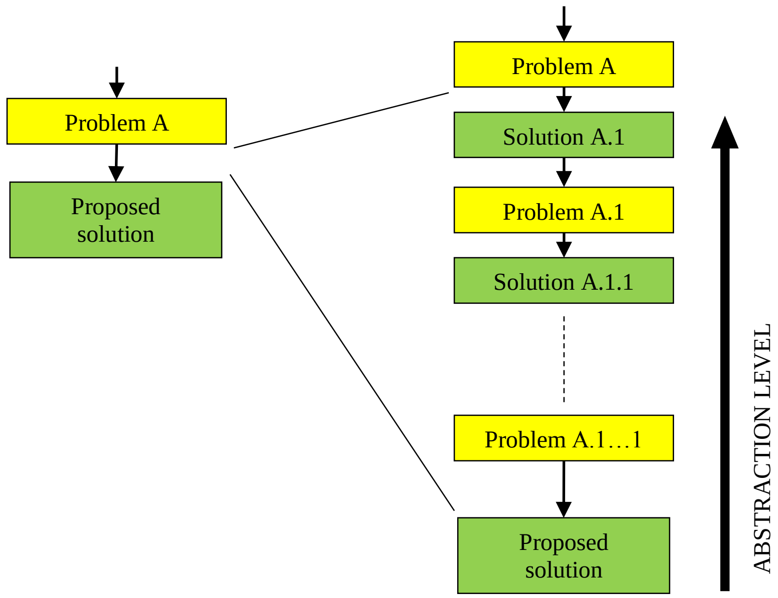


Figure 35. Schematic representation of a generic case where the proposed solution belongs to more extended families of solutions at different levels of abstraction.

Independency of PSN branches

A fundamental rule to be considered when composing the PSN, is that each branch must be developed independently from the others. In other words, problems involving the interaction between solutions belonging to different branches are not stored in the network. It does not means that interaction problems between solutions are neglected in the proposed approach, but only means that they are not considered during the first phase of the conceptual design, i.e. the concept generation. Problem concerning the form of the solutions or combination problems like incompatibilities between solutions of different branches will be faced during the concept composition phase.

However, the importance of this rule is strictly related to the risk of uncontrolled vertical expansions of the net. Indeed, proceeding with A-S-E process recursively, one can theoretically starts from the overall problem formulation, up to the definition of detailed solutions like bearings, springs, screws etc. In this case the PSN becomes unmanageable and then unusable. But the “independency” rule allows to stop vertically the network expansion, since preliminary tests proved that beyond a certain level of detail it is impossible to formulate problems related to partial solutions without knowing anything about the rest of the system.

A simple example can be used in order to show how the rule influences the realization of the PSN, and for that purpose a very simplified conceptual design task is considered, i.e. the conception of a linear actuator (Figure 36). It is worth to notice that for the sake of brevity it has been considered only a “unique” ramification, intentionally neglecting the development of other possible solutions at the various levels of abstraction.

In the example of Figure 36 the problems which are not considered during the concept generation phase, and then which have not to be represented in the PSN, are those highlighted with a red dashed border. Indeed, trying to solve such problems implies to make assumptions about the rest of the system, e.g. “how to connect the spring to the system” implies to choose if the spring is guided in the inner or in the outer diameter. However, this is only a detail which does not change the working principle, but is strictly dependent on the form that will be assumed by the guide (which is the generic solution of the other principal branch, “How to ensure linear motion?”).

For those concerning the “vertical stop” of the PSN, it is possible to observe how the solution at the lowest level network (i.e. the “lever”) generates other three problems, concerning details about its form, which however depends on how the rest of the system has been shaped. These problems are not represented in the PSN. However it is not possible to formulate any other problem about the “lever”, and then the related branch vertically stops.

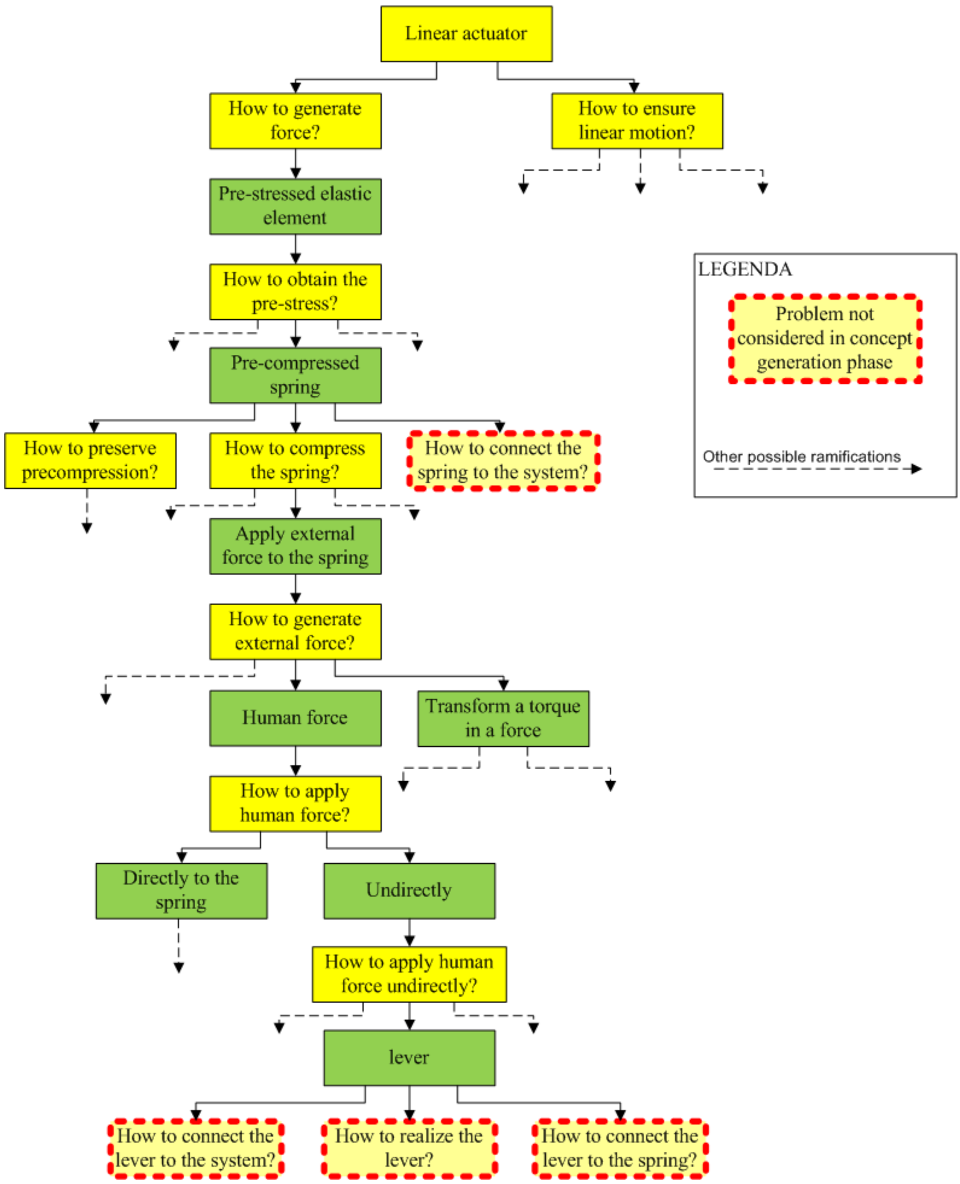


Figure 36. Generic example of a PSN branch where some problems not considered in the concept generation phase have been highlighted with red dashed borders.

Completeness of the PSN

The above introduced independency rule is used to stop the vertical expansion of the net. However, it doesn't mean that the designer activity stops once the development of one or few branches reaches the vertical limit. Indeed, each problem obtained with the "Principal functional problems formulation" rule leads to at least one complete branch of the network, i.e., a branch ending with a solution. This means that all the principal functionalities of the product have been investigated and a set of potential solutions has been inferred. Conversely, neglecting one or more main functionalities leads to an incomplete product concept development. Therefore, such a part of the rule is fundamental. Moreover, in order to ensure the consideration of the complete set of hypothesis stored in the network, the completeness rule comprises another part. Such a part is related to the information gathering activities noted into the PSN by the specific boxes shown in Figure 29. More specifically, a PSN cannot be considered complete until the information gathering has been completed. It is worth to notice that it doesn't mean that all the information must be necessarily acquired. Indeed, there are many possible reasons which can lead the designer or the design team to not gather some information (e.g. secured information, limited resources, etc.). In such a case, the branches of PSN whose information is not complete cannot be considered for the concept composition, because the lack of sufficient knowledge makes problem analysis and related solutions totally unreliable.

6.2.5. Proposed concept composition approach

Once a satisfactory development of PSN has been reached, many possible and different overall concepts may be obtained, which are then represented with sketches or rough CAD models. As shown in Figure 29, any problem, belonging to any level of detail or abstraction, may have more than one possible solution, generating itself different sub-problems and then other partial solutions. Similarly to the decision tree of Marples (1961), in order to manage the composition of a variety of overall solution concepts, the PSN itself may be used as a selection management tool for selecting most promising branches and discarding others. However, such a selection must be performed observing two simple rules. The first rule concerns the main functionalities of the product, i.e. the selection of the branches must be performed in order to avoid incomplete concepts. The second rule is devoted to avoid the adoption of incomplete branches. Indeed, during the development of the network many reasons may lead to unsolved problems, implying the existence of "dead" branches which cannot be used in the concept composition phase. An example of branch selection is reported in Figure 37.

The example is a simple and illustrative PSN, composed by a very small number of branches. In such a case, where the considered problem-solution sequences are highlighted by thicker arrows, it can be observed that only one other possible selection is possible, i.e. solution 1.1 instead of the 1.2. Indeed, in the same picture it is possible to observe that the branch of the main functional problem 1.2 leads to two different solutions, i.e. the number two and the number three. Solution number three cannot be considered as a possible candidate, since it generates one unsolved problem, leading to a dead branch. Differently, solution number two leads to two problems (Pb 1.2.1 and Pb 1.2.2), which however have been solved with solutions 2.1 and 2.2 respectively. Then, for concept composition, both these two solutions must be considered, because they solve the two distinct problems generated by the parent solution number 2.

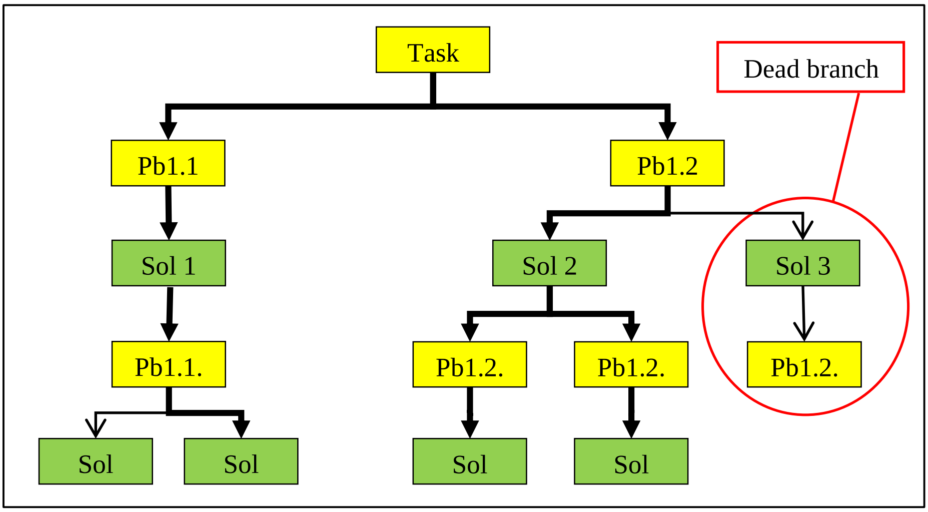


Figure 37. Example of a concept composition by means of the PSN. The chosen branches are highlighted by thicker arrows, and also a dead branch is present.

Selecting some branches and discarding others substantially means to obtain a “reduced” PSN, where a unique problem-solution sequence is possible. However, since it is almost always possible to find more than one acceptable problem-solution sequence in the net, it is often possible to obtain more than one reduced PSN. Therefore, for each of them, there is the possibility to realize many different sketches representing first rough forms of the concept physical structure, to submit to the concept selection phase.

However, in performing the composition of the concept, some compatibility problems may arise among the solutions belonging to different branches of PSN. So, before obtain the sketch of the concept, the recalled integration problems should be solved by the designer, who can take advantage from the well-known Brainstorming, or even by formulating and facing contradictions with tools belonging to the TRIZ body of knowledge (Altshuller, 1984).

Actually, such combination problems are those that lead the designer to consider or to neglect a reduced PSN. More precisely, those reduced PSN which imply hard problems concerning combination and/or implementation, are not considered for the concept combination phase.

6.3. How to obtain modular solutions with the new structured conceptual design approach.

The new approach for conceptual design presented in the previous subsection, beyond the peculiarities which makes it an alternative to classical processes, can be modified to allow generating modular solutions. More precisely, modularity can be faced during the two first phases of the conceptual design process, i.e. the concept generation and the concept combination. Here in the following these two add-ons of the new approach are described in detail.

6.3.1.Modularity during concept generation activities

In order to achieve modularity during concept generation activities, the proposal described in SubSection 6.2 has been taken into account. However, in order to implement such logic in the PSN approach, some modifications have to be performed. More precisely, instead of comparing requirements with the modularity benefits, the problems composing the network have been considered. Then, by using the definition of "modular problem" reported in Sub-Section 6.1, it is possible to identify problems that can be solved with modular solutions. This identification can be performed during the realization of the network, by modifying the ASE logic. Indeed, it simply means to add a further question in the "Analysis" group of the ASE logic, as shown in Figure 38 and Table 6 (Question M1).

Then, once a problem is identified as "modular" the generation of a modular solution starts, as shown in Figure 38, where a specific step is inserted (M2). Obviously, such a step has to follow the same rules of the ASE logic, and then is linked also to the Information Gathering group. However, since the proposed approach does not aim at a mere increment of modularity, the possibility to generate other non-modular solutions remains. Indeed, in this way modular solutions are considered only a group of possible alternatives to solve specific problem, which however could be solved also with other solutions characterized by diverse architectures.

In particular, the generation of the modular solution follows the instructions of the preliminary proposal shown in Sub-Section 6.2, where the standard modularity characteristics are used as design catalogue to inspire the designer. The procedure can be represented in the PSN as shown in Figure 39, where the problem identified as "modular" is schematized by a red-edged yellow box. A hypothetical "module" is represented by the red-edged green box, which automatically generates three problems related to the identification of the three standard groups of characteristics. Then, each of them is solved by taking inspiration from the possible variants of characteristics belonging to the specified group.

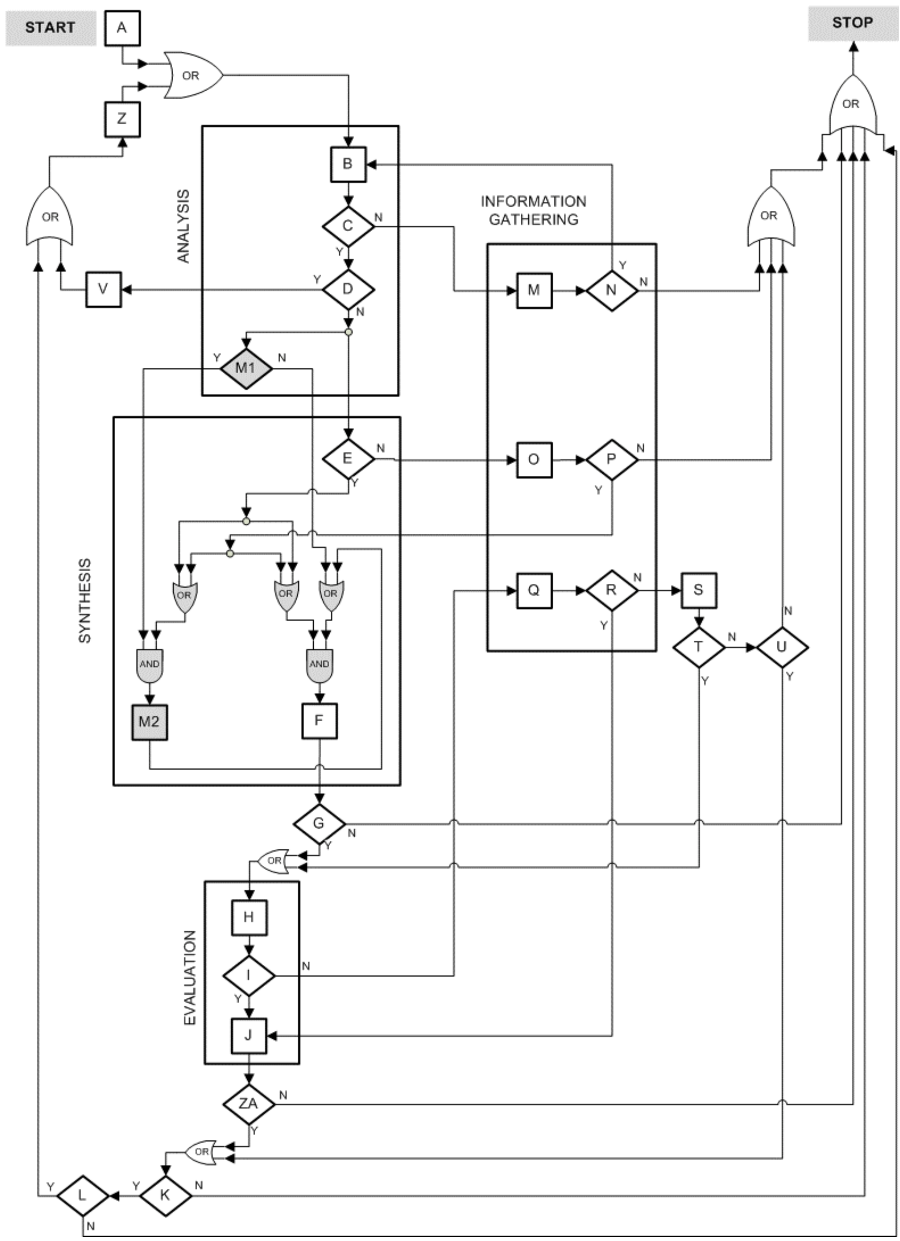


Figure 38. The ASE logic modified for conceiving modular solutions (new boxes in grey).

Table 6. New steps in the ASE logic, for conceiving modular solutions.

M1	Is the problem a modular one?
M2	Try to generate modular solutions.

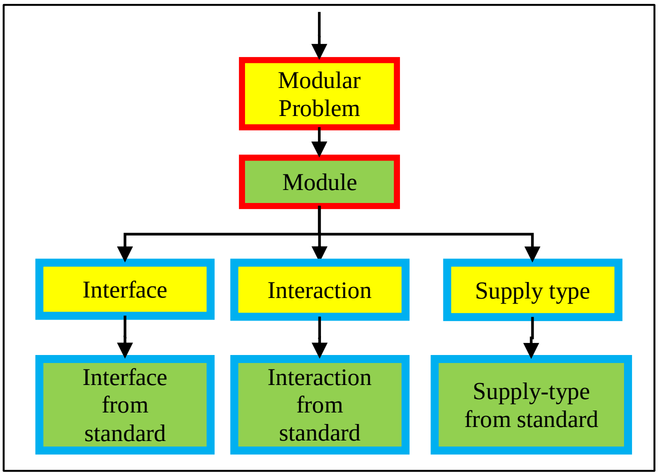


Figure 39. Module generation on the PSN.

6.3.2.Modularity during concept composition activities

Preliminary tests showed that the modularity approach described above is not capable of taking into account all the possible chances of generating modularity. Indeed, some of the requirements of the design task cannot be considered in the network, due to the application of the independence rule. As a matter of the fact, requirements like “ease of assembly” are often strictly related to the form, the number of parts and the way they are connected. But the independency rule avoids to consider such problems into the PSN. Nevertheless, as reported in Table 1, “ease of assembly” is one of the modularity benefits and then it is simple to deduce that a modular architecture can give aid in fulfilling the requirement.

Therefore, since the formalisms introduced in the modified ASE logic cannot be used without the formulation of a problem, when facing modularity issues outside the PSN, it is necessary to adopt a different way to proceed.

Fortunately, beyond the identification of modular problems, as demonstrated in Sub-Section 6.1 it is also possible to identify “modular requirements”, i.e. requirements that can take advantage from modularity in their fulfillment. Then, a similar identification procedure can be pursued when composing and subsequently sketching the concept, i.e. analyzing the requirement list, looking for requirements not faced into the PSN, and compatible with modularity benefits.

Once such a kind of requirements has been identified, the designer can evaluate three different possibilities:

- Develop the concept as a module to be connected to an existent super-system.

- Develop some modules internal to the system (Figure 40).
- Proceed without considering modularity.

The differences among the first two possibilities (the first and the second in the previous bulleted list) reside in the "granularity level" (Chiriac et al. 2011) considered for the generation of modularity. More precisely, the first chance concerns the development of the system considering it as a module to be connected to other parts of an existent system. In such a case the designer has to conceive the overall concept taking inspiration from the modularity types shown in Table 2, i.e. those concerning the interface, the interaction and the supply type.

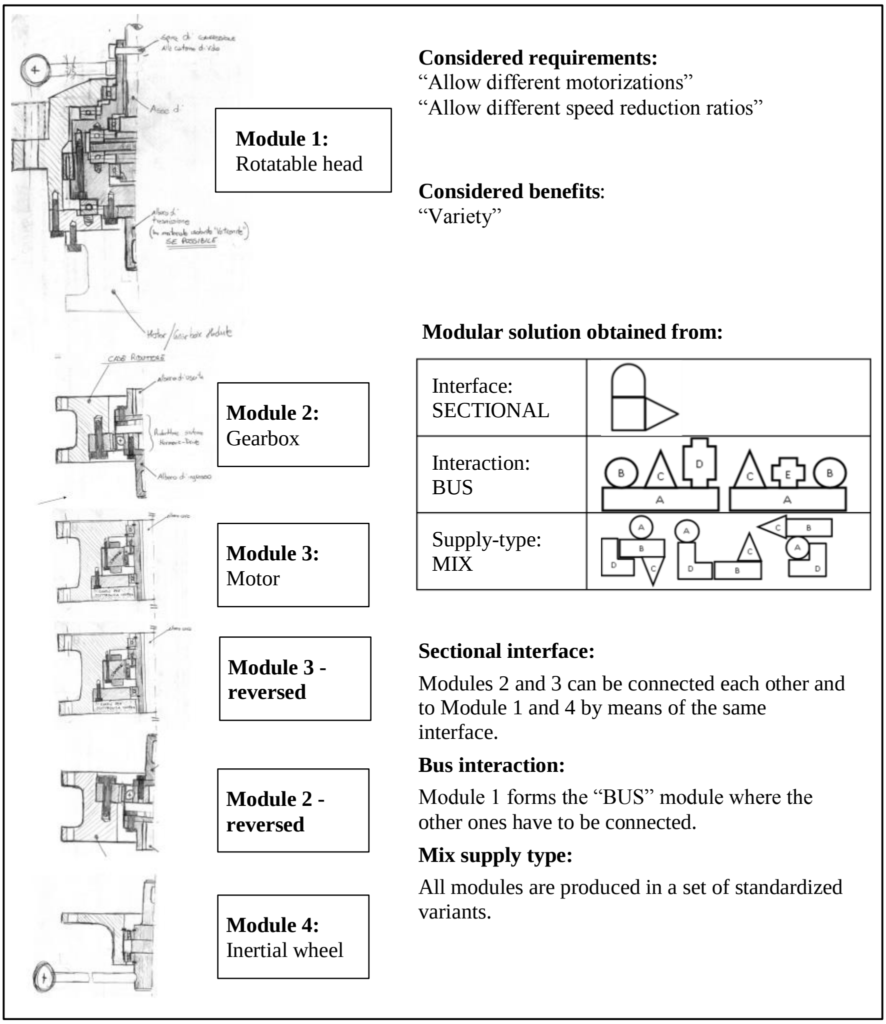


Figure 40. Example of a concept composed by different modules.

Moreover, if the internal components needs to directly interact with the super-system, the related interfaces have to be coincident with the interfaces that overall concept shares with the super-system. In other words, it correspond to the application of the "modular concept generation" process introduced in Sub-Section 6.1, applied at an higher level of granularity (i.e. the overall system to be designed is a component of a super-system).

Differently, in the second case the designer has to think about some internal modules fulfilling specific functions. In this case the granularity level coincides with the system and the modular concept generation process introduced in Sub-Section 6.1 can be applied. Figure 40 shows an example where internal modules have been conceived in order to fulfill two specific requirements compatible with modularity benefits. In such a case, standard modularity types have to be considered for determining assemblies composing the first granularity level of the system.

References

Altshuller, G. S. 1984. Creativity as an exact science: The Theory of the Solution of Inventive Problems. Gordon and Breach Science Publishers.

Andreasen, M. M. 1980. "Machine Design Methods Based on a SystematicApproach – Contribution to a Design Theory." PhD diss. (In Danish), Department of Machine Design, Lund Institute of Technology, Sweden.

Aurisicchio, M., and Bracewell, R. 2013. "Capturing an integrated design information space with a diagram-based approach." Journal of Engineering Design 24 (6): 397–428. doi:10.1080/09544828.2012.757693

Bonnema, G. M., and Van Houten, F. J. A. M. 2006. "Use of models in conceptual design." Journal of Engineering Design 17 (6). 549–562. doi:10.1080/09544820600664994.

Chiriac N., Holtta-Otto K., Lysy D., Suk Suh (2011) E. Level of Modularity and Different Levels of System Granularity. Journal of Mechanical Design, vol. 133, no. 10.

Cross, N. 2000. Engineering design Methods_Strategies for Product Design. 3rd ed. J. Wiley.

Dorst, K. (2006). Design Problems and Design Paradoxes. Design Studies, 22(3), 4–17.

Evbuomwan, N. F. O., Sivaloganathan, S., and Jebb, A. 1996. "A survey of design philosophies, models, methods and systems." Journal of Engineering Manufacture. 210 (4): 301–320.

Fiorineschi, L., Frillici, F. S., and Rissone P. 2011. "A comparison of Classical TRIZ and OTSM-TRIZ in dealing with complex problems." In: Proceedings of the Triz Future Conference 2011: Systematically Innovating for Sustainable Competitiveness. Edited by G. Cascini and T. Vaneker, 103-112. Dublin: Institute of Technology Tallaght.

Fixson S.K. (2003) The Multiple Faces of Modularity: A Literature Analysis of Product Concept for Assembled Hardware Products. Technical Report 03-05, Industrial and Operation Engineering, University of Michigan

Jones, J. C., and Thornley, D. G. 1962. Conference on design methods. London: Pergamon Press.

Hatchuel, A. (2002). Towards Design Theory and expandable rationality : The unfinished program of Herbert Simon . 1 From Decision making to Design theory : Journal of Management and Governance, 5(3-4), 260 – 273.

Khomenko, N., De Guio, R., Lelait, L., Kaikov, I., () "A framework for OTSM TRIZ-based computer support to be used in complex problem management", International Journal of Computer Applications in Technology Vol 30, Issue ½, 2007.

Malmqvist, J. 1997. "Improved function-means trees by inclusion of design history information." Journal of Engineering Design 8 (2): 107-117. doi: 10.1080/09544829708907955.

Marples, D. L. 1961. "The decisions of engineering design". In: IRE Transactions on Engineering Management EM-8 (2): 55-71.

Ulrich, K.T., Eppinger, S.D. , "Product Design and Development", 4th ed. McGraw-Hill, Irwin, 2003.

Watts, R. D. 1966. "The elements of design." In: The design method, edited by S.A. Gregory, 85-95, London: Butterworths.

7. Performed investigations and tests.

Here in this section a comprehensive description is reported about the investigations and tests performed in order to achieve preliminary validations of the proposals introduced in Section 6. The activities have been mainly performed in academic environment, involving the research group and some engineering students.

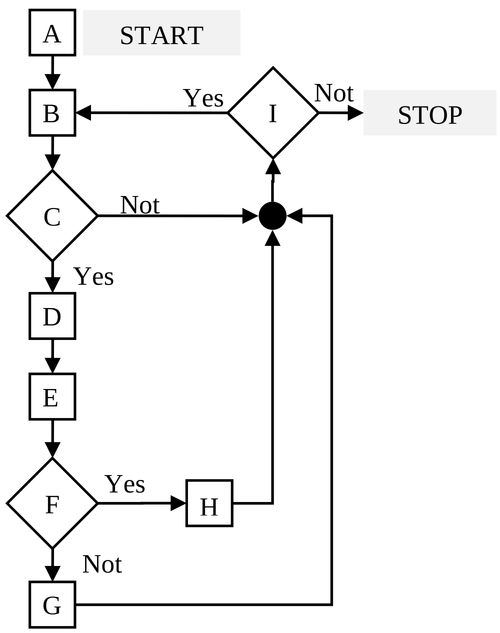
7.1. Performing a first investigation on the rise of modularity

As introduced in Section 6, the objective of this investigation consists in gathering information on how modularity arises in the design process. It involves the necessity of analyzing many design processes carried-out by designers and the related outcomes. Substantially, there is the need to identify the adopted technical solutions, assess them for finding modular characteristics, and somehow go back to the reasons that led the designers to adopt these characteristics. Such a process requires the mapping of the solutions that appear during the conceptual design activity and the linking of these ones to the original problems. To this purpose, the Network of Problem (NOP), derived from the OTSM-TRIZ base of knowledge (Khomenko et al. 2007), constitutes is a valid tool since it allows visualizing relationships between problems and solutions belonging to different level of detail of the system. This is the reason why a problem-solution approach has been chosen and the two definitions of modular problem and modular solutions introduced in Section 6 have been adopted.

Coming down in performing the investigation, a not negligible problem to be faced is represented by the need of collecting and managing a big amount of data related to real case studies whose design processes, requirements and outcomes must be well known. For this scope, two possible solutions arise, i.e. the observation of design processes in real time and the analysis of already performed design tasks. The first chance privileges the completeness and exhaustivity of the data, but direct observations may involve too time to obtain the required amount of information. The analysis of already performed design processes potentially involves a minor amount of time resources for its implementation, however there are some important drawbacks to be considered. First of all, the success of this approach is strongly dependent on the completeness of the information that can be gathered from the sample of case studies. Furthermore, there is the need to verify and ensure that the design intent was not explicitly oriented towards the search for modular solutions, otherwise the results of the analysis miss the meaning. A way to solve this problem is to consider only design processes where the designer or the design team were not learned about modularity. Moreover, there is the necessity to relate modular problems with the corresponding solutions but, since the design process cannot be directly observed, the relationships between problems and solutions must be reconstructed using a sort of "reverse engineering" approach. The coevolving path involving problems and solutions can be recreated by interviewing the designers that have carried out the examined case studies. To the scope of this work, it has been decided to adopt the analysis of already performed design tasks as way of investigation, thanks to the availability of three case studies whose design processes are sufficiently characterized in terms of requirements, outcomes and main design problems faced by designers in the concept development phase. It is worth to notice that every design problem related to the identification of the functions, the physical principles and the basic forms of a part of the product, are considered here as belonging to the conceptual design phase. The considered sample of convenience is sufficient to show how the investigation method works, as well as to obtain preliminary outcomes to be discussed. However, the same method can be adapted and subsequently adopted also for investigations performed through the direct observation of the design activity.

7.1.1.Investigation approach

The proposed investigation method is constituted by four main activities that are here described in detail, while the logic of the suggested approach is shown in Figure 41.



A	Retrieve the conceptual design problems list
B	Compare one problem with the modularity benefits
C	Is the problem compatible with modularity benefits?
D	Identify the related solution
E	Compare the solution with the modularity types
F	Is the solution associable to a modularity type?
G	Assign null score
H	Assign unitary score
I	Are other conceptual design problems still present?

Figure 41. Logic of the design processes investigation approach.

The first step consists in the acquisition of design problems which characterize the design process under investigation. For that purpose, designers involved in the investigated project are asked about the followed design process and faced problems. The main design problems are typically related to the fulfillment of the functional requirements of the system, while the others are related to more detailed aspects. Once the list of design problems has been obtained, the subsequent step is the identification of the modular problems by performing the comparison with the modularity benefits introduced in Section 3. This is a fundamental activity, since it allows the emerging of the linkage between informal occurrence of modularity and specific modularity benefits. This step is carried out with the presence of the designers involved in the examined case study. In this way it is possible to avoid eventual misunderstanding due to the use of improper or incomplete descriptions of the problems. To give an example, for the design of a biomass grinder, a problem encountered by designers was: "How to allow to process different raw materials?". Only after a confrontation, the modularity benefits which fit with this problem definition have been identified in "Variety (j)" and "Customization (k)". Indeed, both the benefits are related to the diversification of the product model, although the first suggests the use of standardized parts while the second considers the use of custom-made ones.

Conversely, design problems like "how to increase reliability" and "how to reduce energy consumption" do not match with any of the benefits, so they have not been considered in the analysis since, they can be considered as non-modular problems. This is in accordance with the literature since performances aspects of the product are optimized moving towards integrality (Ulrich 1995), (Hölttä-Otto and De Weck 2007).

The third step of the method is the retrieval of the solutions adopted to solve the modular problems identified in the previous step. As for Step 2, the active participation of the designers that have carried out the activity is required to perform the task. Subsequently, the identification of modular solutions, among the retrieved solutions, is performed by searching for decoupled interfaces and module characteristics belonging to the three groups defined in Section 3. In order to show how the modular solution identification is performed, one of the investigated cases is considered and a description of the process is reported in the following. The solution (Figure 42) belongs to the design process related to an innovative biomass grinder, where the considered modular problem is that previously mentioned in this Section, i.e. "How to allow to process different raw materials".

Referring to the example of Figure 42 and considering the rotor assembly as the "system", the results of the comparison with the modularity characteristics are explained in the following:

- **Interface type:** the modularity type is "SLOT", because each component has a different interface with the rotor disc. This solution resembles the definition of "Slot Modularity" reported in Section 3, i.e. "all the interfaces are of different type". It can be observed that the interfaces are not decoupled, in fact, a variation of the internal diameter of the inner rotor cage implies a modification on the plate. The same for the other components, but for other diameters.
- **Interaction type:** the modularity type is "Swapping" because different components can be interchanged in the same rotor assembly. It is in fact equivalent to the definition of "Swapping Modularity" reported in Section 3, i.e. "two or more components can be interchanged in a module in order to create product variants". Moreover, in the adopted solution there is a "Bus" component,

i.e. the rotor plate, to which the others are connected, allowing to obtain rotor variants by changing the part version or by eliminating the outer cage.

- **Supply type:** the modularity type can be considered congruent with the definitions of both the "Fabricated-to-Fit" and the "Mix" modularity given in Section 3, because rotor cages can be both standardized parts (internally to the firm) or custom made parts.

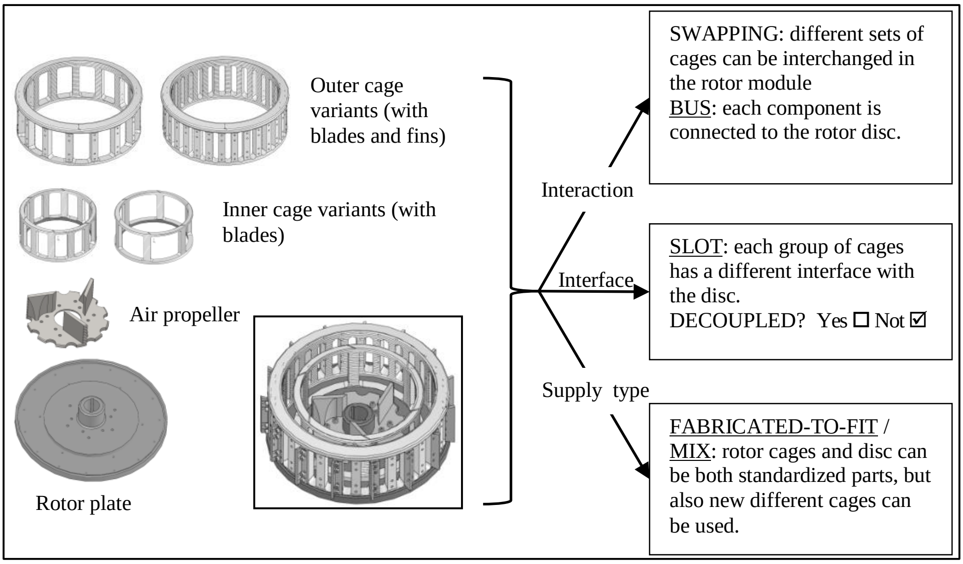


Figure 42. Identification of the modularity characteristics involved in the solution related to the problem: "How to allow to process different raw materials?"

Eventually, the last step of the method consists in the assignment of a binary score (1 or 0) in order to use the results to evaluate how modular solutions are related to modular problems. More specifically, a unitary score is given to each modularity benefit whenever it is involved in the identification of a modular problem that is associated to a modular solution. Conversely, a null score is given to the benefit when the related solution is not modular. Taking again as a reference the example of Figure 42, even three modular characteristics have been found then, given the definitions of Section 3, it has been possible to consider the solution as modular, and consequently a unitary score is assigned to both the benefits involved in the identification of the related modular problem.

7.1.2.Considered case studies

The case studies chosen for the investigation are a system to grind wet biomass, a platform for performing stratospheric ballooning experiments (Gondola) and a hydraulic pole driver for excavator's heads. It is worth to highlight that the three considered cases concern the development of experimental prototypes, where no explicit intent to obtain modularity were considered.

The project of the biomass grinder was originally born to improve the wood pellet manufacturing process (Cascini et al. 2008), trying to introduce a new technology capable to eliminate some shortcomings in the current wood grinding systems which fail when they handle wet raw materials. In order to develop such a system, a design activity was engaged with the aim to develop a prototype of a totally new system for performing experimental activities (Figure 43).

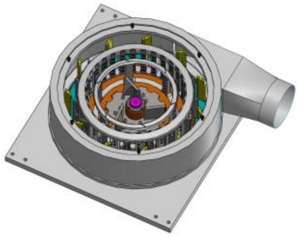


Figure 43. Excerpt of the biomass grinder CAD model

The second project concerns the design of an innovative platform to support the devices for performing stratospheric experiments by using probe balloons. The design task was focused on the search of a new solution aimed at reducing flight costs (Boscaleri et al. 2009). A schematic embodiment of the solution is represented in Figure 44.

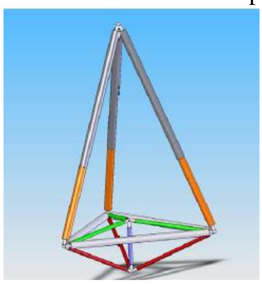


Figure 44. Schematic CAD model of the Gondola structure.

Eventually, the last case study consists in a design activity aimed at developing a pole driver system prototype for excavators heads (Figure 45), capable of preserving the integrity of the wooden poles since in the current systems some problems emerge during the process.

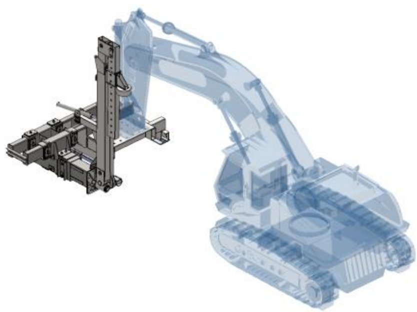


Figure 45. Hydraulic pole driver mounted on a excavator head

It is worth to highlight that due to the restricted amount of data considered in this work, it was not possible to obtain a statistical reliability of the results. However they have been successfully used to develop and evaluate the potentialities of the proposed investigation method and, furthermore, they allowed to rise some not negligible research questions.

As shown in Table 7, the set of design problems considered for the investigation are those which, in their formulation, present the possibility to be solved with modularity. In the sample it has been found that 14 out of 18 considered problems were effectively solved with a modular solution. More in detail, and considering that sometimes more than one benefit may be involved in the identification of a modular problem, the outcomes of the test are reported in Figure 46. That graph, developed considering the scoring results, shows the comparison between the occurrence of modularity benefits involved in the identification of modular problems, and the number of times in which the related modular problem was solved with a modular solution.

Table 7. Modular problems and related benefits for the considered case studies

Case	Design problem	Associated Modularity benefit
Biomass grinder	How to ease the maintenance of the cutting elements?	Ease of maintenance.
	How to allow to process different raw materials?	Customization; Variety.
	How to allow different output size of the processed material	Customization; Variety.
	How to allow upgrades of the cutting elements	Allow upgrades/part changes .
	How to allow to test different impact blades configurations	Customization.
Stratospheric platform	How to ease of multiuser management	Design team management.
	How to reduce the design costs of the gondola	Design reuse.
	How to increase the reuse the gondola after landing?	Component reuse.
	How to ease the transportation and recovery operations?	Disassembly time.
	How to ease of the assembly process?	Ease of assembly.
	How to obtain different shapes for the same gondola?	Variety; Customization.
	How to allow different positions of the Pivot axis?	Variety; Customization.
	How to reduce manufacturing costs	Economy of scale.
Pole driver	How to allow compatibility with different crane heads?	Customization.
	How to allow compatibility with different poles?	Customization.
	How to ease the maintenance of sliding parts?	Ease of maintenance.
	How to allow upgrades?	Allow upgrades/part changes .
	How to split the project into two distinct sub-tasks?	Parallel development.

7.1.3. Investigation results

Despite the limited amount of data, the results depicted in Figure 46 show the existence of a link between the informal occurrence of modularity and certain types of design problems. However, the way to confirm such a kind of evidences is to consider a more extended sample of design processes. In reference to the scope of the investigation, such preliminary results are encouraging and suggest that the proposed method can be used to verify the informal occurrence of modularity during conceptual design and to obtain information to investigate on the mechanisms that leads to this phenomenon.

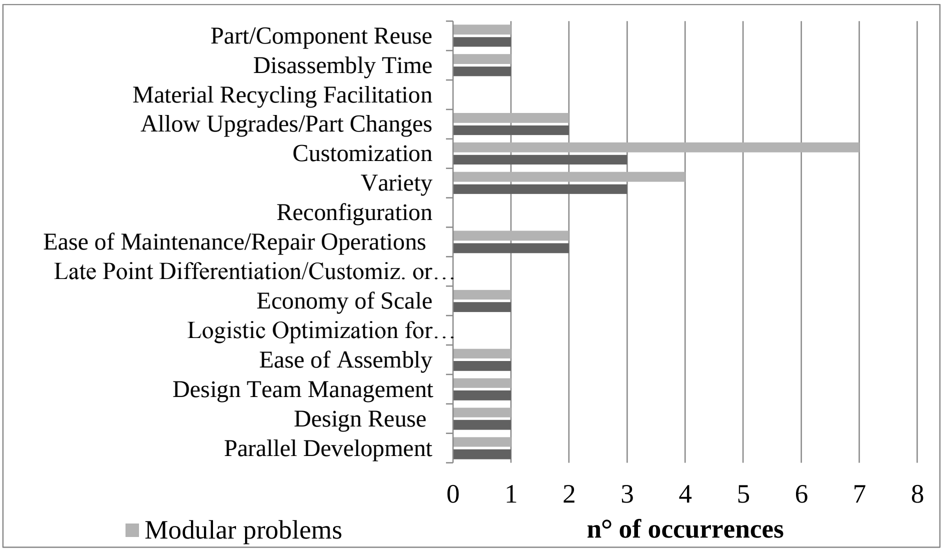


Figure 46. Comparison between the occurrence of modularity benefits used for identification of modular problems, and the number of times in which the related modular problem was solved with a modular solution.

Moreover, the identification of the modular solutions also allows to highlight the modular characteristics that were the objectives of the solution development. For instance, it has been found that for all the three times where "Customization" and the "Variety" benefits were involved in the identification of the modular problems related to the modular solutions (Figure 46), the "Swapping" interaction type was indicated as the driver that guided the designer. This means that the proposed method also allows to investigate on the existence of a direct relationship between each modularity benefits and specific modular characteristics.

Furthermore, it can be noticed that in four of the eighteen modular problems, none of the modularity characteristics was observed in the adopted solution. In these cases another coincidence can be observed, i.e. the modularity benefit involved in the identification of the modular problems was the "Customization" one (in one case together with "Variety"). What stated above means that the proposed method could also allow to investigate when different types of non-modular solutions can be used as a valid alternative to modular ones.

Looking to the modular solutions of the considered sample, only an half of them presents modularity characteristics belonging to all the three categories. For instance, in the solution of Figure 42 the implemented interface between the rotor plate and the other components is not decoupled. The analysis of the solutions performed together with the designers during the application of the proposed method highlighted that, a posteriori, the adoption of a decoupled interface would have been preferable. In fact, the interfaces between the rotor plate and the other components were thought only to ensure the correct positioning, neglecting the explicit need to obtain different configurations of the rotor and to allow drastic modifications of the prototype. This is a case in which, because of the lack of a specific support in the concept design phase, the designer attained a wrong or, at least, an incomplete result in the development of a required modular solution. Although this evidence cannot represent a proof due to the limited number of design process that have been analyzed, this kind of outcomes suggest that further researches are needed to develop design methods and tools capable to guide the designer in the identification of suitable modular solutions whereas modular problems arise. However it is worth to notice that in any case, also in presence of a modular problem, modularity has to be considered only as "potential" solution. So the last decision concerning the choice of the best solution must always be performed by considering the set of product requirements.

7.2. Test of the modularity approach for non-structured conceptual design processes

The present test aims at evaluating the approach introduced in Sub-Section 6.1, in particular concerning the generation of modular solutions in early non-structured conceptual design activities. More precisely the test aims at verifying the applicability of the proposal and its actual efficacy in reaching modular concepts.

The test consisted in administering a case study regarding the design of a product starting from a limited set of given requirements, to a sample of convenience composed by eighty-two engineering students. The expected outcome was the sketch of a solution and a short description of it. The considered sample of student was randomly subdivided into the following two groups:

- Analysis Group (AG). Formed by forty students equipped by a specific material developed to support them in using the proposed approach.
- Control Group (CG). Formed by forty-two students, left completely free in the development of the concept. Such a group constitutes a reference in order to evaluate if the developed approach, used by the AG group, is capable of generating impacting results in terms of modularity.

A common case study has been assigned to both the groups, concerning the concept development of a multifunctional and customizable pen. More precisely, the functions to be implemented were the pencil, the pen and the eraser. Two main requirements to be satisfied by solutions, i.e. "multifunction" and "customization", have been considered since they can be linked to modularity benefits. Indeed, the fulfilment of the "multifunction" requirement can take advantage by modularity, as expressed by the benefit

"Reconfiguration/Flexibility in use" introduced in Table 1. Similarly, the "customization" requirement is linked to modularity through the "Customization" and "Variety" benefits.

However, other requirements have been furnished in order to frame the design task. Indeed, for what concerns the class of the product, an intermediate level from medium to luxury pens has been given as a reference. While, concerning ease of use, maneuverability of standard pens and absence of additional tools for using it, were given as expected characteristics of the final product.

Eventually, it is also worth to notice that beyond the absence of any training about modularity before the test, students were not learned about any type of systematic conceptual design process.

Here in the following, detailed descriptions are reported about the supporting material given to the AG and the metrics used to validate the results.

7.2.1.Supporting material for the analysis group

As for the CG, the AG was equipped with a description of the two main requirements and a description of the form in which the results were expected to be delivered. Also a pre-printed sheet to be used for drafting and describing the solution was distributed to each student of the two groups.

However, since the AG was supposed to follow the experimental approach for the fulfilment of the two main requirements, additional instructions were furnished. More precisely, the following set of information and supporting material has been supplied:

- Explicit request to follow the approach to fulfil the two main requirements
- Detailed instructions for a correct use of the process shown in Figure 27 (Section 6).
- Three tables containing a simplified description and a generic example concerning each modularity type (Table 8). Each table contains modularity types as grouped in Table 2.

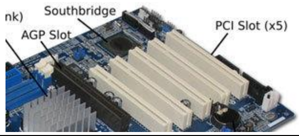
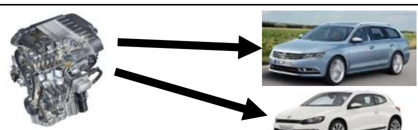
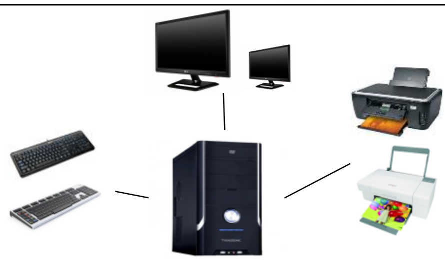
7.2.2.Evaluation of the concepts

The results of the test were assessed according to the following criteria:

- Degree of fulfilment of the two main requirements. In order to assess how much the concept matches the design task.
- Degrees of Feasibility and usability of the solution. In order to purge the sample from unfeasible and/or unusable solutions.
- Modularity level. In order to assess modularity of the concepts, and to evaluate the impact of the proposal.

Each level composing the above mentioned metrics corresponds to a numeric value. The generated solutions have been ranked through a score assigned according to the level of satisfaction of each criterion.

Table 8. The three modularity tables furnished to the AG

INTERFACE GROUP	Slot modularity	Each module type has a different interface.	
	Bus modularity	All modules have the same interface with a bus component.	
	Sectional modularity	Modules can be connected each other by mean of the same interface.	
INTERACTION	Swapping modularity	Different modules can be interchanged each other.	
	Sharing modularity	The same module is shared by different product variants.	
	Bus modularity	A base module can be connected to different other modules.	
SUPPLY TYPE	Mix modularity	The system is formed by standard modules.	
	Cut-to-fit modularity	The system is formed by standard modules and customizable ones.	

Fulfilment of the two main requirements

In order to evaluate how well the concepts fulfil the main requirements, two specific metrics have been formulated (Table 9). By means of them, author aims not only at a mere ranking of the concepts, but also at evaluating the reliability of the non-structured approach. Indeed, by observing the relationship between modularity level and the main requirement fulfilment, it is possible to observe the actual link between modularity and related benefits, obviously only for the specific case study.

Table 9. Metrics and score (S) used to evaluate the fulfilment of the two main requirements.

S	Multifunction	Customization
4	Possibility to use the three functions without any need of assembly or disassembly	Possibility to completely change the aspect by changing also specific aesthetic components
3	Possibility to implement only two functions concurrently. The third function obtainable as alternative of one of the others, by means of single connection or disconnection.	Possibility to completely change the aspect by changing components aimed at implementing the three functions, by means of a simple connection or disconnection.
2	Possibility to implement only two functions concurrently. The third function obtainable as alternative of one of the others, by means of more complex operations.	Possibility to partially (completely) change the aspect by changing components aimed at implementing the three functions, by means of a simple connection or disconnection (by mean of complex operations).
1	Not possible to implement Eraser and Pencil concurrently.	Not possible to change the aspect of the system

Feasibility and usability of the concepts

Due to the limited time and the skill of the students involved in the test, author foresee the possibility to obtain conceptual solutions affected by feasibility or usability problems. For this reason a specific metric has been developed in order to rank the delivered concepts in terms of their potential feasibility and ease of use. A rough scale composed by three levels has been considered in order to rank the concepts, i.e. not feasible (score = 1), existence of some doubts concerning feasibility or usability (score = 2), no feasibility or usability problems detected (score = 3).

Three levels have been considered in order to obtain three different classes of concepts, but the principal objective of the application of the metric is to discard those concepts which shows evident unfeasibility or non-usability. The third class has been created in order to isolate those concepts that do not show any evident problematic, and then to allow the possibility of an investigation on such a restricted sample.

Modularity level of the concepts

This metric aims at assessing the modularity level of the solutions developed by students, through the observation of the structures represented in the concept sketch. As already said, many definitions of modularity can be found in literature, belonging to different engineering domains and based on different perspectives (Salvador, 2007). Some of these definitions may differ when using terms like, module, chunk, component and element. Then, in order to avoid ambiguity and supply a reference for the scope of this work, the following key concepts have been considered, which are based on the consideration that different levels of detail can be identified in a product:

- System: Every part or assembly belonging to a determined level of detail may falls under this definition. At the highest level of detail the system corresponds to the product.
- Component: With this term any physical element is identified, intended as single part or assembly which constitutes the system at the succeeding level of detail.
- Module: It is intended here as a particular component connected to the rest of the system and which is identifiable with the modularity definitions given in Section 3.

Literature offers many types of modularity metrics (Hölttä-otto et al., 2012), however many of them consider a number of details that are not available in the concept sketches produced by students.

For such a reason, an elementary modularity measure has been considered here to formulate the levels for ranking students outcomes. More precisely, one of the two metrics proposed by Mattson and Magleby (2001), as reported by Hölttä-otto et al. (2012), has been taken into account. In particular, only the number of functions and modules are considered to asses modularity:

$$\text{Modularity} = \frac{\text{number of modules}}{\text{number of functions}} \tag{1}$$

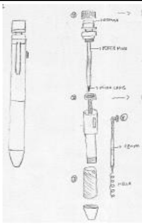
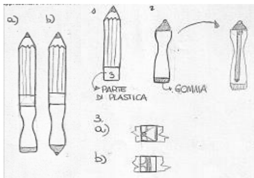
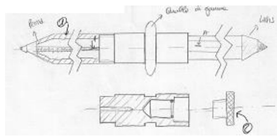
Taking (1) as reference, the following three levels of modularity have been formulated for the specific case study of the present test:

- The pen, the pencil and the eraser functionalities are implemented by components that cannot be used independently from the rest of the system (score = 1).
- Only one of the three requested functions is implemented by a component that can be used independently from the whole system (score = 2).
- All the three requested functions are implemented by distinct components that can be used independently from the rest of the system (score = 3).
- The above mentioned metric is intended to be applied considering the whole product as "System".

Here in the following, an evaluation is shown of the concepts produced by students belonging to both the groups, according to the above described criteria.

Concepts ranked as "unfeasible" have been discarded, reducing the sample from eighty-two to fifty-eight persons (thirty-one for the control group and twenty-seven for the analysis group). Such a not negligible reduction is probably due to the limited time and to the level of expertise characterizing the sample, i.e. beginners. Three examples of solutions conceived by students, independently from the group, are shown in Table 10. It is worth of noting that students also produced a written description of each concept, by means of which the scores have been assigned.

Table 10. Three examples of concepts and the related scores.

Sketches of the concepts			
Modularity	1	2	3
Feasibility	2	3	3
Multifunction	4	3	4
Customization	2	3	4

7.2.3.Effect of the approach on the concept modularity level

By comparing the scores assigned to the concepts belonging to CG and AG groups, a not negligible difference arises. Indeed, considering the mean values obtained for each group, the AG is characterized by a higher score (+16,8%). Such a value is even doubled if only concepts characterized by the highest level of feasibility are considered (+33%) (Figure 47). However, it is worth of noting that if considering only such a level, the sample is reduced to fifteen and twelve people, respectively for CG and AG.

In order to assess the statistical reliability of the observed difference, a two-sample t-test (Sheskin 2003) has been executed on the modularity mean values of the two groups. In the first case (feasibility scores two and three), comparing the modularity mean values of AG and CG leads to a p-value of 0,076 (Figure 48). On the contrary, by repeating the t-test considering only most feasible solutions, the p-value becomes 0,005 (Figure 49). Taking 0,05 as reference p-value to accept the alternative hypothesis (mean values are different), it is possible to observe that in the first case the p-value is slightly out of limits, but this only means that the null hypothesis (mean values are statistically equal) can be discarded only with a slightly higher uncertainty value. Differently, in the second case the difference between the two mean values is validated by the test with a confidence level of almost one hundred percent.

The difference between the two groups in terms of modularity of the concepts can be considered a first evidence of a not negligible impact of the proposed approach.

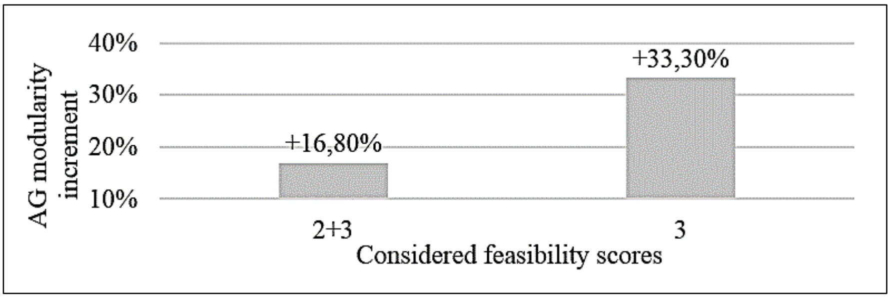


Figure 47. Increment of the mean modularity score observed in the AG. The percentage has been calculated taking as a reference the values of the control group.

Two-Sample T-Test and CI: modularity; Group

```
Two-sample T for modularity

Group   N   Mean  StDev  SE Mean
1      31  1,871  0,806     0,14
2      27  2,185  0,834     0,16


Difference = mu (1) - mu (2)
Estimate for difference:  -0,314
95% upper bound for difference:  0,047
T-Test of difference = 0 (vs <): T-Value = -1,45  P-Value = 0,076  DF = 54
```

Figure 48. Two-sample t-test executed for feasibility levels 2 and 3.

Two-Sample T-Test and CI: modularity; Group

```
Two-sample T for modularity

Group   N   Mean  StDev  SE Mean
1      15  2,000  0,756     0,20
2      12  2,667  0,492     0,14


Difference = mu (1) - mu (2)
Estimate for difference:  -0,667
95% upper bound for difference:  -0,254
T-Test of difference = 0 (vs <): T-Value = -2,76  P-Value = 0,005  DF = 24
```

Figure 49. Two-sample t-test executed for feasibility level 3

7.2.4.Link between modularity and the two considered benefits

Considering only the requirement satisfaction metrics, it is not possible to find significant differences between the mean values of AG and CG. However, also CG students develop modular concepts, confirming again that modularity may arise informally during the conceptual design phase (Ulrich and Eppinger, 2007).

Here in the following, the effect of modularity on the satisfaction of the main requirements is evaluated on the whole sample, i.e. composed by AG and CG together, and on AG and CG separately. First of all, Chi-squared (X^2) tests of independence (Sheskin 2003) has been performed on modularity against the satisfaction level of the two main requirements. Indeed, such a kind of test is capable to assess the probability of association or independence of two variables, analysed through the considered sample of data. The critical X^2 has been obtained for the degrees of freedom characterizing the variables (six), and a probability level of 0,05. If the X^2 calculated for two variables is lower than the critical one, then it is not possible to reject the null hypothesis, i.e. the absence of any relationship. Instead, if the calculated X^2 is higher than the critical one, it is possible to assert that a relationship between the two variables exists. Table 11 shows the results of the test performed between modularity and the satisfaction level of the two requirements. It is possible to observe that both the two variables are related to modularity when considering the whole sample. Instead, if considering the groups separately, it has not been possible to reject the null hypothesis for modularity vs customization.

Table 11. Results of the X^2 independence tests performed on the sample

TEST	X^2 (AG+CG)	X^2 for AG	X^2 for CG	Critical X^2
Modularity VS Multifunction	32,56	18,32	14,57	12,59
Modularity VS Customization	13,49	12,45	3,34	

However, even when it is possible to observe a relationship with modularity, the X^2 independence test does not describe how the variables are interrelated. In order to estimate the trend, the mean values of each of the two variables have been calculated for each modularity level. Results are shown in Figures 50 and 51, respectively for the whole sample and the separated groups. In Figure 50 it is possible to observe different behaviors of requirements satisfaction in relation to the considered modularity level of concepts. Moreover, for what concerns modularity vs. multifunction, a difference can be observed also between the two groups (Figure 51). The reason of such a difference is not clear, but the presence of concepts characterized by an intermediate level of feasibility may introduce some noise. However, by considering only the best concepts in terms of feasibility (level 3), two not negligible problems arise, i.e. the previously mentioned reduction of the sample, and the absence of data at the lowest level of modularity for the AG. For that reasons, such an incomplete set of results have not been considered here.

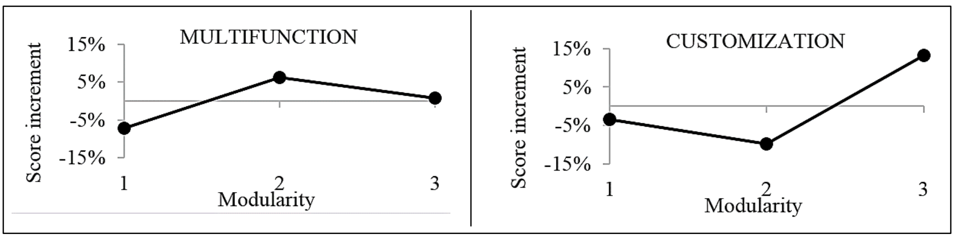


Figure 50. Influence of modularity on the main requirements satisfaction. The values represent the difference among mean scores calculated for concepts characterized by specific modularity levels, and the global mean scores, i.e. 2,78 for Multifunction and 2,34 for Customization.

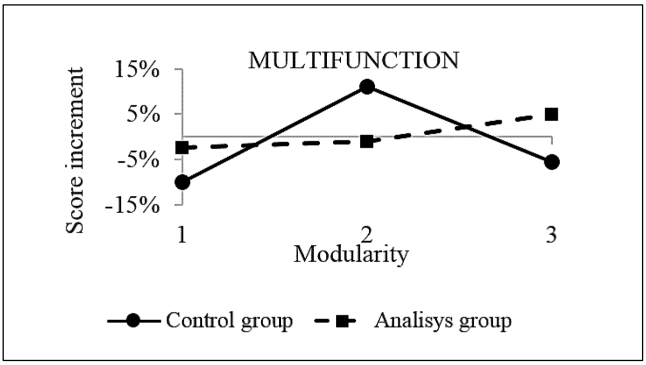


Figure 51. Differences observed between the AG and the CG, about the influence of modularity on the Multifunction requirement satisfaction. The global mean scores are 2,78 for Multifunction and 2,34 for Customization.

7.2.5.Considerations about results

The results substantially report that the proposed experimental approach is capable of producing a not negligible effect in terms of modularity level of the concepts. Moreover, the evaluation of the effect of modularity on the main requirements satisfaction confirms the existence of a relationship between modularity and the considered benefits, however, it has been shown that it is not linear. Such a statement is in accord with similar observations perfomed by scholars on other case studies and other benefits (Collado-ruiz and Capuz-rizo, 2013). Thanks to these promising results, this first experience forms the basis for future research activities devoted to the development of an upgraded version, which aims at being integrated in structured conceptual design processes.

7.3. Evaluating the PSN approach

The present sub-section describes a first evaluation of the PSN approach, i.e. the new approach proposed to assist conceptual design processes. It is not a proper test, indeed the activity can be subdivided into the following two parts:

- A first comparison with the conceptual design model considered as a reference in this thesis. Such an activity aims at highlighting commonalities and differences of the proposed approach with the Pahl and Beitz one.
- Observation on first applications of the method. Here the objective is to verify the applicability of the proposal and to gather information for the development of a first guideline for designers.

Such parts are described in detail here as follows.

7.3.1. Literature case study: a comparison with the Pahl and Beitz approach.

The proposed concept generation approach is here compared with the classic functional decomposition and morphology procedure, by means of a literature case study application. The considered example refers to the concept development of a household one-handed mixing tap, which is one of those used in Pahl et al. (2007) to show how their approach works. It is worth to put beforehand that the following comparison is not aimed at criticizing the results reached by Pahl and Beitz for the specific case study. Indeed, it is impossible for authors to be aware of all the information, the issues and any other instance that they faced during the real design process. Differently, the comparison here presented aims at demonstrating that the proposed approach can be used in order to overcome the lacks of the functional decomposition and morphology approaches. Thus, a short description of the original case study application is reported, highlighting the most important requirements and assumptions affecting the results shown in the mentioned book. Then, the application of the proposed approach is shown, describing the process in detail. At the end of the section, a discussion is reported, in order to evaluate the differences between the two approaches.

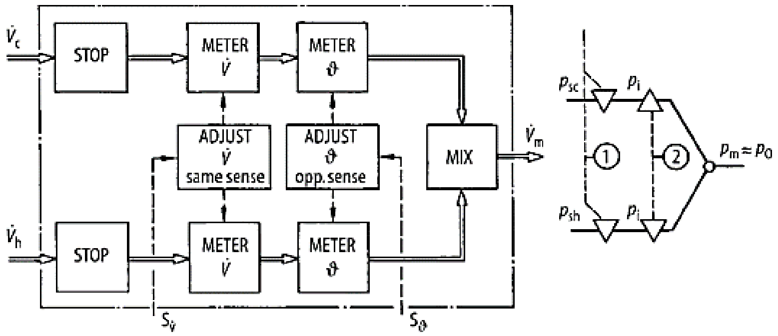
The household one-handed mixing tap case study: short description.

The considered case study concerns the development of a water mixing tap for domestic use, starting from a list of product requirements. A short excerpt of such a list, reporting the most relevant specifications, is reported here in the following:

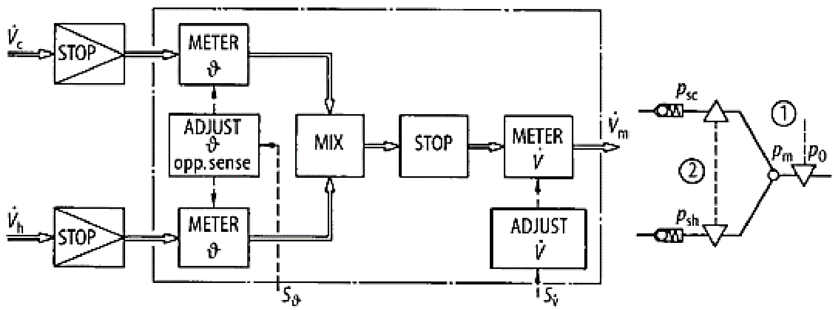
- The temperature of the output flows must remain unchanged after regulating the volume flow rate
- The volume flow rate must remain unchanged after regulating the output temperature
- The use of external energy is not allowed
- Obvious operations, simple and convenient handling

Furthermore, Pahl and Beitz make some relevant assumptions concerning the selection of the principles to be used for implementing the fundamental functions of the system. In order to regulate the flows, the “valve” or “diaphragm” physical principle are adopted because of their worth proved in previous products of the company from which the case study originated.

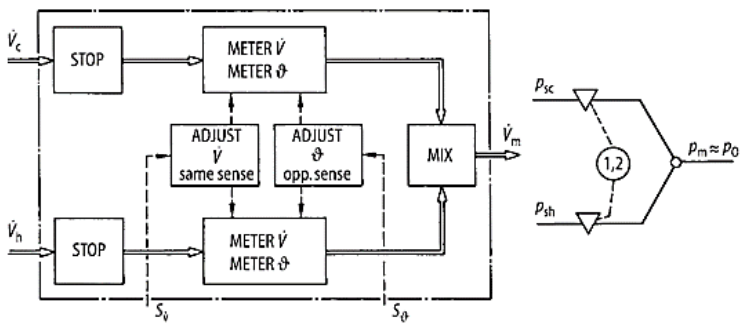
Before generating the functional structures of the solution variants, relationships between flows have been established on the base of the requirements. Thus, they obtained a mathematical relationship between volume flow rates, flow pressures, temperatures and loss characteristic of the valve. After that, they realize three distinct EMS functional models corresponding respectively to three concepts differing in the stopping, regulating and mixing sequences (Figure 52).



Functional model 1



Functional model 2



Functional model 3

Figure 52. The three functional models of the one-handed mixing tap, elaborated by Pahl and Beitz in their book (from Pahl et al. 2007).

The first two functional models contemplate the concurrent regulation of the cold flow rate (Vc) and the hot flow rate (Vh) in order to control the output temperature (Tm) and flow rate (Vm). The first consists in regulating Vc and Vh in two steps before mixing. Conversely, the second one considers a regulation of the flows before mixing, in order to regulate Tm, while Vm is controlled after mixing. The two solutions are quite intuitive at the level of detail considered by Pahl et al., and their behavior has been tested in terms of linearity of regulations. It is not clear how they reached such kind of information at this development level, however some graphs have been reported, showing the regulations behavior at different pressure conditions. This has been done also for the third solution, which has been considered as the most suitable one, thanks to the linear characteristic of the temperature regulation. But the steps which led to such a kind of solution are neither clear nor trivial to be performed by the designer, since a different type of "functional" box has been introduced, with the aim of obtaining better performances of the thermo-hydraulic model. Indeed, while in the first two solutions, the functional model was composed by boxes with single functions, the third one presents boxes with two functions (see Figure 53 b).

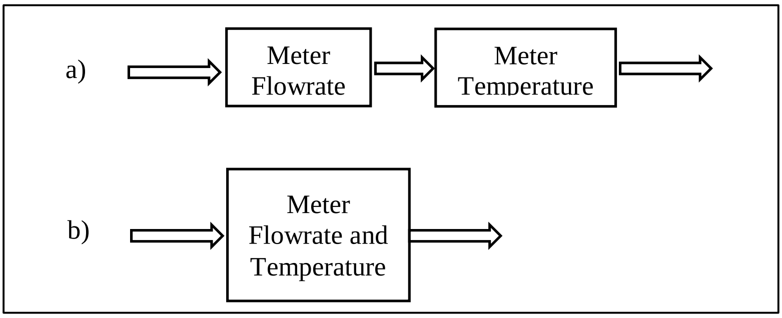


Figure 53. The functional model of the third solution (b) merges two functions into a single box, in order to obtain better performances of the product.

Here it is not discussed the correctness of the use of such a type of functional approach, but the attention is focused on how a designer can be able to jump from the first two solutions to the third one. Indeed, it seems that such a kind of solution has been obtained after the evaluation of the hydraulic behavior of the flow regulation systems, leading the designer at avoiding the presence of two valves in sequence, and then, at implementing the two functions into a single component. However, the process continues with the selection of the best functional model, followed by a brainstorming process performed to find a variety of working principles for the solution implementation. Subsequently, most promising ideas have been selected and used for creating solution variants expressed in form of sketches.

The PSN approach applied to the mixing tap case study.

Before starting with the application of the proposed approach, it is important to observe that concerning the case study, only the information reported in the book (Pahl et al. 2007) were considered for the development of the problem-solution sequences. It means that the "information gathering" activities have not been performed and then, the ASE steps concerning information (as represented in Figure 27) are not considered in this example. Consequently, also the information tracking peculiarity of the PSN is not showed in this section.

The conceptual design process starts from the analysis of the requirement list and the considerations about the original case study. In this way it has been possible to formulate the main design task, to decompose it into the principal sub-problems and then to start with the ASE process for constructing the PSN (Figure 54, Table 12 and Table 13). Here in the following it is described how the network has been realized.

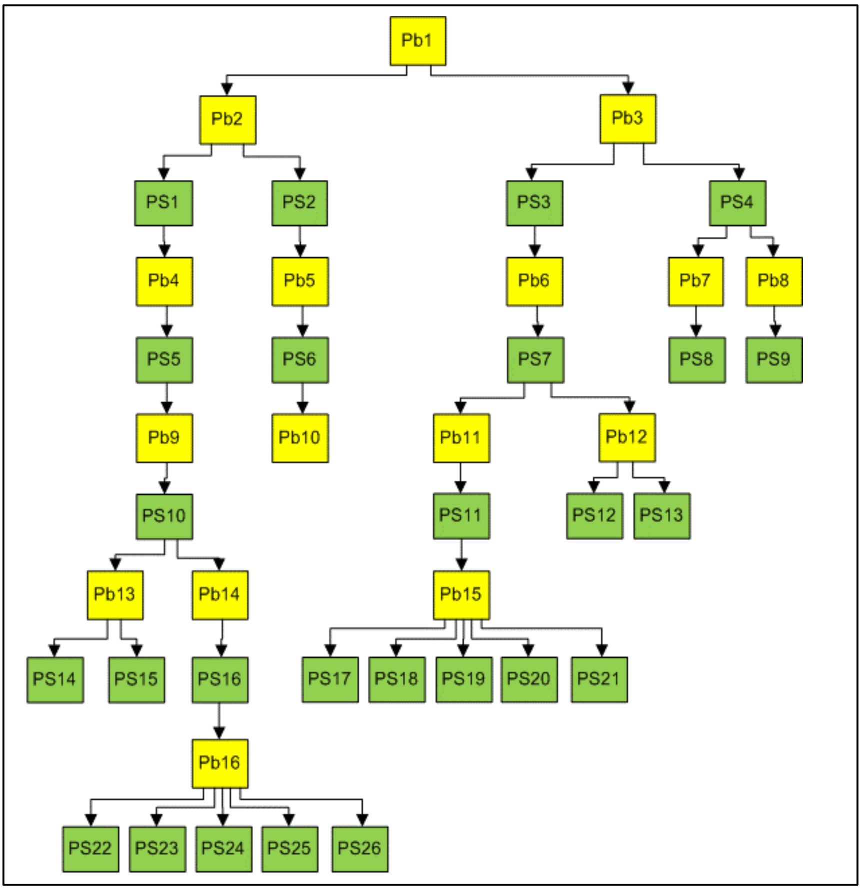


Figure 54. The PSN for the one handed mixing tap case study.

As foreseen by the overall process schema (Figure 27) and considering the first rule of the proposed set (see Sub-Section 6.2), the main design task has been formulated at the maximum level of abstraction required for the present case study. Indeed, by observing the requirement list of the original case study it is quite obvious that the designer has to develop a "New household one-handed mixing tap", starting from the white sheet.

So considering the simplified schema of Figure 27, the next step concerns the identification of the main solution independent design problem at the highest level of abstraction. Then, the two main functions have been identified and expressed in form of problems to be solved, according to the rule reported in Sub-Section 6.2. At the highest level of abstraction, the desired functions concern the regulation of the output temperature from the minimum to the maximum level, and the regulation of the output flow-rate from zero to the max allowed value. According to the above mentioned rule, nothing has been expressed concerning "how" such functions can be implemented.

Table 12. Problems (Pb) of the PSN shown in Figure 54.

n°	Description
Pb1	New household one-handed mixing tap
Pb2	How to regulate the output temperature?
Pb3	How to regulate the total flowrate?
Pb4	How to prevent total flowrate changes?
Pb5	How to prevent total flowrate changes?
Pb6	How to prevent output temperature changes?
Pb7	How to prevent backward fluxes when the mixed flow is stopped?
Pb8	How to regulate the total flow?
Pb9	How to keep constant the sum of the hot and cold flowrates?
Pb10	How to keep constant the sum of the hot and cold flowrates?
Pb11	How to perform the regulation?
Pb12	How to perform the regulations concurrently?
Pb13	How to perform opposite regulations?
Pb14	How to realize flow regulation?
Pb15	How to regulate the flow area?
Pb16	How to regulate the flow area?

Thus, the ASE process has been applied to the defined problems (Pb2 and Pb3 on Figure 54, explained in Table 12). In both cases, the problem has been analyzed, verifying the possibility to understand it in a complete manner. Since the available information was considered sufficient, the succeeding step was to investigate on the possibility of a further problem decomposition, independently from solutions. In each case, such a possibility was not found, because any attempt led to inconsistent problem formulations. For example, splitting the problem Pb3 into "How to stop single flows?" and "How to regulate single flows", implies the consideration of two solutions, i.e. stopping the single flows before mixing, and regulating the total flow-rate by controlling single flows. Another possibility was, e.g., to split the problem Pb3 into "How to stop the total flow?" and "How to regulate the total flow?". But this kind of decomposition is useless and redundant, since the case study considers the adoption of the well know "valve" or "membrane" working principle. This implies that there is no reason to justify the regulation and the stop as two distinct functions, because it is well acknowledged that a valve can regulate the flow-rate from its maximum value to zero, and then stop it.

At this point, the ASE process prescribes to verify if the owned know-how is sufficient to try to solve the considered problem. As stated before, the data listed in the book of Pahl et al. (2007) were considered sufficient. Then, it is the time to find solutions, but beyond the consideration of the ASE logic, the concept generation method proposed here implies to consider also the PSN rules. In this case, the considered rule is that expressed in Sub-Section 6.2.4 (i.e. “Follow the correct sequence of abstraction levels”), concerning the correctness of the abstraction level of the proposed solutions. Considering for example the problem Pb2 (How to regulate the output temperature?), two possible solutions were proposed (see Figure 54 and Table 13), i.e. “Regulate both flows concurrently with the same movement before mixing” and “Regulate both flows with independent movements before mixing”. According to the above mentioned PSN rule, these are the two possible “families” of solutions, compatible with the main requirement, i.e. the use of a single hand. It has not been possible to find any other solution at a higher level of abstraction nor at the same one.

Table 13. Solutions (PS) of the PSN shown in Figure 54.

n°	Description
PS 1	Regulate both flows concurrently with the same movement before mixing
PS 2	Regulate both flows with independent movements before mixing
PS 3	Regulate both flows before mixing
PS 4	Regulate the total flow after mixing
PS 5	The sum of the two regulated flows must remain constant
PS 6	The sum of the two regulated flows must remain constant
PS 7	Same and concurrent regulation for both flows (constant ratio)
PS 8	Use unidirectional valves for the two tubes
PS 9	Use a standard valve
PS 10	Opposite and equal regulations of the two flows
PS 11	Flow area regulation
PS 12	Enveloped wire
PS 13	Rigid connection
PS 14	Rigid connection for translation
PS 15	Rigid connection for rotation
PS 16	Flow area regulation
PS 17	Adjustable cone top
PS 18	Adjustable spherical top
PS 19	Sliding plate
PS 20	Rotatable bored sphere
PS 21	Chamfered cylinder
PS 22	Adjustable cone top
PS 23	Adjustable spherical top
PS 24	Sliding plate
PS 25	Rotatable bored sphere
PS 26	Chamfered cylinder

Then, following the ASE process, it is now time of the evaluation step. The formulated solutions result easy to understand, and the list of requirements reported in the original case study was used to extract the evaluation parameters. It is worth of notice that the evaluation of the partial solutions reported in the PSN does not need to be operated by any kind of tool, but simply by the designer himself or herself. In this case, due to the high level of abstraction, in order to accept the solutions it is sufficient to verify that there is not any contradiction with the expressed requirements. Then, once the solutions have been accepted, the first ASE process stops for this specific branch. However the overall process does not stop, because problems related to the proposed partial solutions (PS1 and PS2 in Figure 54) have to be formulated. According to the rule expressed in Sub-Section 6.2, the correct sequence of abstraction levels has to be ensured also when formulating problems, and considering the requirement concerning the independency of flow and temperature regulation, the problems Pb4 and Pb5 were formulated. Both the problems can be expressed in “How to prevent total flow-rate changes?”, but according to the PSN rule concerning the independency of the branches (Sub-Section 6.2) a distinct problem box has been realized for each solution. Then, as explained in Section 6, the ASE process can start again from this second level, leading to other solutions and the related problems, and so forth.

According to the above mentioned independency rule (see Section 6), the vertical expansion stops when a “critical” detail level is reached, and then it is no longer possible to formulate problems without combining solutions. Indeed, considering the present case study, it is possible to ascertain that formulating a problem for the solutions proposed at the end of each branch, implies to know information about the choices performed on other branches. For example, if considering the solution PS13 (rigid connection), related to the problem Pb12 (How to perform the regulation concurrently?), a logical path would lead to formulate problems related to its physical realization. But the physical realization can be sensibly different depending on the solution chosen for solving Pb15 or Pb16 (How to regulate the flow area?). Indeed, for a classical bored sphere (PS20 and PS25) such a connection would be able to transmit a torque, while if considering sliding plates (PS19 and PS24), a force is needed. The final PSN rule (see Sub-Section 6.2) implies that this kind of approach is applied to all the main branches of the network. The same rule also implies that at least a complete branch must be reached and all the information-gathering processes have been concluded.

For this specific case study application, solutions belonging to the lowest levels of the branches have been identified without any specific attempt of being creative or innovative, but only a sample of quite intuitive ones has been considered with the sole aim of completing the net. However, as well as for the original case study, any kind of creative methods for finding solutions would be applied. In particular, it is possible to consider the application of problem solving tools, as for example those belonging to the TRIZ base of knowledge (Altshuller 1984), or even all the methods listed in Pahl et al (2007) or Ullman (2010). But it is not in the scope of this case study application to reach a complete final solution, or even to describe how to implement the results into concept sketches or embodiments.

After the specification above, it is now possible to simulate a concept composition aimed at obtaining the three results considered in Pahl et al. (2007). According to what expressed in Section 5 concerning concept composition, in Figure 55 the reduced PSNs related to the three solutions are reported. It is worth to remember to readers that for the above expressed reasons, the PSNs have been considered only at a partial detail level, neglecting low level problems and solutions.

By observing Figure 55, a first evidence appears, i.e. only two reduced PSNs have been obtained for representing the three solutions of the original case study. To explain such a fact, it is important to note that the two problem-solution sequences differ in the selection of the solution used to solve the problem Pb3 (How to regulate the total flow-rate?). The selection used for Figure 55(a), is the solution PS3 (Regulate both flows before mixing), while in Figure 55(b) the considered solution is the PS4 (Regulate the total flow after mixing).

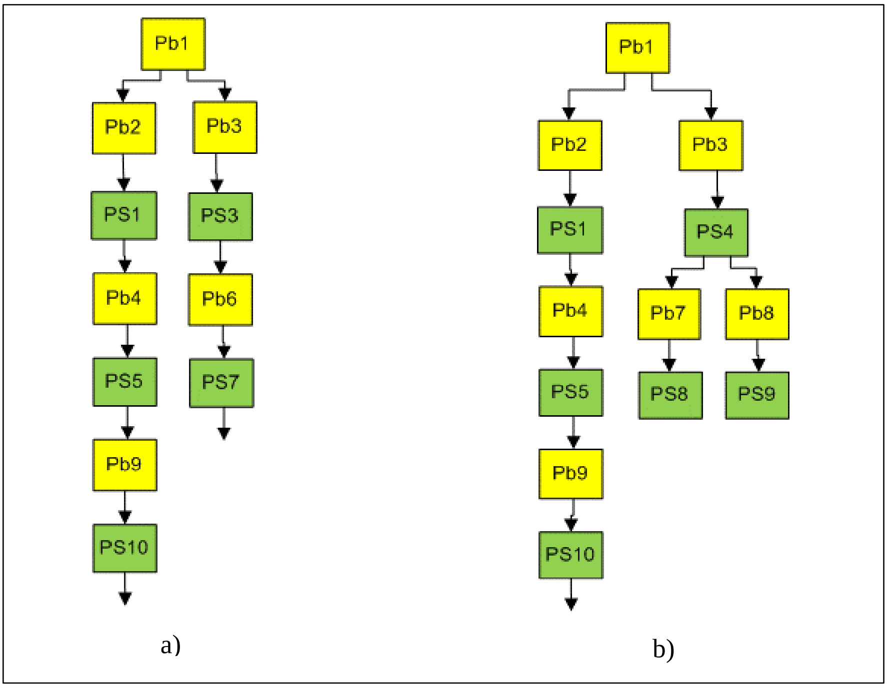


Figure 55. Reduced PSNs for Pahl et al. case study's functional solutions (see Tables 7 and 8 for contents).

The first and the third solution of the original case study both belong to the first reduced PSN, because both of them regulate the two flows before mixing. There is no way to differentiate the two concepts at this level, since the problem derived from PS3, is Pb6 (How to prevent output temperature changes?) is common to both of them, as well as the related solution PS7 (Same and concurrent regulation for both flows). Even considering further levels of the PSN is not useful to differentiate the two solutions, because they concern the development of details which can be used in both type of concepts.

Such an observation can be considered a confirmation of the doubts introduced at the beginning of this section, concerning the development of the third solution by Pahl et al. (2007). Indeed, it is now evident that such a solution can be considered as an improvement of the first one, and not a simple alternative. Therefore, it is possible to infer that in order to develop a solution in which a new type of valve is devoted to flow and temperature regulations, the first two more intuitive solutions have to be somehow tested. Only in this way, it is possible to identify that in order to reach linearity for regulations, the presence of two valves in series must be avoided. Actually, another possibility exists, i.e. to reach the required information "before" composing the concept, but in this case, the development of the first two solutions has not sense.

Similarities and differences between the two approaches

Comparing the approach described in Pahl et al. with the PSN one, it is possible to find some similarities. First of all, both start from a requirement list and finish with a set of concepts to be selected. Moreover, the PSN approach starts with a task identification which can be considered very similar to the "task clarification" of Pahl et al. A further similarity is that also the realization of the second level of the network, i.e. the identification of the main functions, is performed considering the system as a black box.

However, the proposed approach is capable of considering any kind of design problem which can be formulated in the form "How to verb noun?". For example, the independency of the two types of regulation is a fundamental requirement but cannot be expressed in terms of functions, and then it results impossible to directly represent it into the EMS functional model. Differently, the PSN allows the formulation of the problem related to such a requirement (i.e. Pb4 in Figure 54, Table 13).

Another important difference concerns the possibility given by the PSN, to represent all the possible concept variants in a single graphical schematization. Such a peculiarity allows a noticeable reduction of the effort if compared with the multiple schematizations needed for functional decomposition and morphology approaches. Indeed, with the network of Figure 54 all the three considered solutions are represented at the same time. The EMS functional model is certainly a more detailed and efficient tool for representing concepts from a pure functional point of view, but this peculiarity leads to the usability problems listed in Section 2. On the contrary, the PSN allows to drastically reduce bias during the concept generation phase by means of a concurrent consideration of any possible (but admissible) solution at different levels of detail. In other words, instead of generating single models, all the potential solutions are summarized in a unique schema and only the most promising combinations are considered for the concept synthesis.

Beyond the differences observed during the concept generation phase, others reside in the concept combination one. As reported in Section 2, morphological approaches foresee the use of tools for mapping functions and solutions, e.g. the morphological charts (Pahl et al. 2007). However, a distinct chart has to be realized for every single functional model considered for implementation. As already stated, this is an onerous task. Conversely, the PSN map is realized only one time, and, as shown in Figure 55, a series of reduced PSNs can be obtained by selecting the considered branches and discarding the others (following the rules expressed in Section 6). Each of them represents a family of possible concepts, sharing the same problem-solution sequences.

7.3.2. First applications of the approach

One of the most important differences between the classical functional approach and PSN has not been shown in the previous section. Such a difference is the possibility of keeping track of the information gathering activities characterizing conceptual design processes. For such a reason, an excerpt of other case studies is shown in this section, in order to give some examples concerning the above mentioned and other peculiarities of PSN. The first case study has been submitted to a sample of twelve mechanical engineering students, at the last year of the course. The second one is an industrial case study, in which PSN has been successfully used in order to develop conceptual solutions.

The automatic door hinge case study

Such an academic case study was though in order to perform a first test of the PSN approach. For this purpose, a sample of twelve engineering students was considered, to which the case study was administered in a form manageable and compatible with the available time resources. Then, a reduced list of requirements to be fulfilled was proposed, where some specific indications concerning the weight of the door, an indicative value of the needed torque, and indications concerning the door closing velocity were furnished. Further requirements were also considered, i.e. the easiness of the installation, the possibility to regulate the closing speed and the aesthetic requirements in terms of forms and position of the hinge.

Before submitting the case study to students, they were trained about the PSN approach by means of specific theoretical lectures totally amounted to six hours. Moreover, they attended at a practical demonstration lasted three hours.

Students rapidly understand the process and start to apply the method in a limited range of time. However, some uncertainties emerged when facing the development of the network, primarily concerning the identification of the right level of abstraction for both problems and solutions formulation. Such a kind of problem, accordingly to Bonnema et al. (2006), can be considered quite normal to non-expert designers. At the end of the test, all the involved students produced their PSNs, so they obtained the reduced ones during the concept combination phase, and generated some sketches about the related conceptual solutions (e.g. Figure 56).

Considering an excerpt of a PSN realized by a student (Figure 57a), it is possible to see how information gathering processes can be represented through the net, allowing to keep track of them. In the specific case, the information gathering concerns the acquisition of more detailed information for evaluating the possibility to use disc springs for generating a force. Differently, in Figure 57b, it is possible to observe how preliminary sketches can be used directly in the PSN as an aid for the quick understanding and the preliminary development of the proposed solutions.

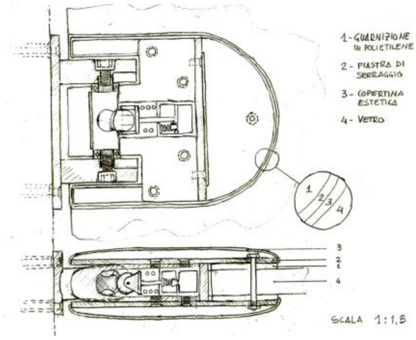


Figure 56. One of the sketches produced by students for the automatic door hinge case study.

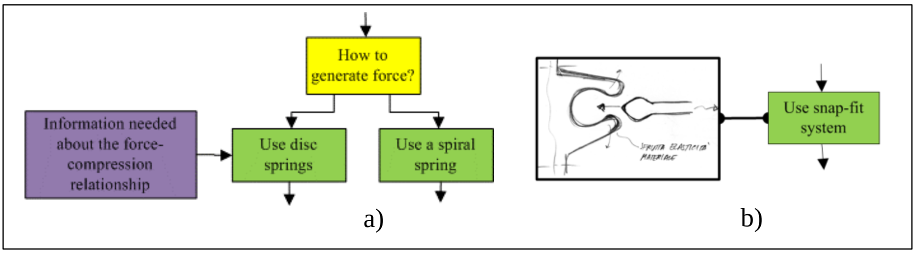


Figure 57. Excerpt of PSNs (translated in English) realized by students, where a information gathering activity (a) and sketches of the partial solutions (b) are represented.

Actually, not all the PSNs realized by students were correct, because beyond the problems related to the abstraction levels, also few mistakes concerning the network representation were committed. The most common error (an half of the sample) is the use of a unique problem box for different parent solutions, leading to a network difficult to be interpreted. However, it is worth to notice that for realizing the network, only the elementary functions of the well-known software Microsoft VISIO ® were used. Then it was quite tedious for students to realize boxes and to connect them. This is the reason that probably led them to use a single problem box for similar problem formulations.

Other minor problems concern problems formulation and solution description, which sometimes where expressed through too short sentences. Problems related to incorrect understanding of the requirements are not considered here, since such a kind of errors is independent from the PSN approach.

<u>The industrial core cutter machine case study</u>

The industrial application of the PSN approach concerns the design of a cutting system for an automatic core cutter machine. Such a design task comes from the design staff of a firm, which ascertains that the complexity of assembly operations for their current product was too high, leading to high costs of the related operations. Since they cannot find a comprehensive solution with their practical design approach, the authors proposed to face the problem through the PSN approach.

A check-list was developed in order to formulate questions for interviewing the firm to acquire a first set of design requirements. Such a check list took inspiration from those proposed by Pahl et al. (2007) for conceptual design, and by Pugh (1991). Subsequently, once the main parameters to be modified were identified, the PSN approach was applied. By following the rules expressed in Section 6 and the ASE process shown in Figure 28, it has been obtained a final network constituted by 116 problems and 185 partial solutions (see Figure 58). The obtained number of problems and related solutions is not negligible; however it depends on the type of design activity to be faced. In other words, the proposed approach may lead to very complex and large networks, but problems and solutions represented in such a graph are nothing else than those generated through the thinking process of the designer. As a matter of the fact, it is well acknowledged that design processes, even if limited to the conceptual phase, involve a large number of variables and parameters to be somehow considered. The PSN only represents them in "problem-solution" form.

Since it was identified that, for what concerns the complexity of assembly operations, the most impacting part of the machine was the cutting system, the design task focused on a complete redesign of such a part. Therefore, the first box of the net was "Design a new cutting system". Then, the main solution-independent problems were those related to the cutting of the core and to the regulation of the cutting length. To each problem, the ASE processes was applied, and by considering the rules expressed in Section 6, the final PSN was obtained. As shown in Figure 58, during the process, many information gathering activities were performed and then booked on the net. Also many sketches of the partial solutions were produced and inserted in the PSN, allowing also a rapid confrontation with the firm's staff in order to evaluate them or to find additional evaluation parameters (as foreseen in the information-gathering activities of the ASE process).

From the obtained network, by performing the concept composition process described in Section 6, six solutions were extracted at a relatively high level of abstraction, representing different ways to perform the cutting. After that, a selection activity based on the screening process proposed by Ulrich and Eppinger (2007) was performed and the most relevant solution was identified. Subsequently, the solution was further developed up to the realization of a detailed CAD model of its embodiment (Figure 59).

The number of the considered conceptual solutions appears very small in relation to the extension of the PSN. For sure, despite the supervision of the authors, the skill of the student involved in the experiment was determinant. Indeed, it is in the opinion of the authors that some partial solutions and the related branches would be rapidly avoided by an expert designer, sensibly reducing the PSN extension. However, in order to evaluate potentialities and lacks of the approach, the student activity was kept more independent as possible. The test showed that also in this case the same problems concerning the choice of the right level of abstraction, faced by students in the door hinge case study, appears again. Moreover, the complexity of the network was relatively high, so it cannot be easily managed through the use of a standard graphical software.

The problems emerged from these experiences, lead the authors to think about some future developments of the work, reported in the following section.

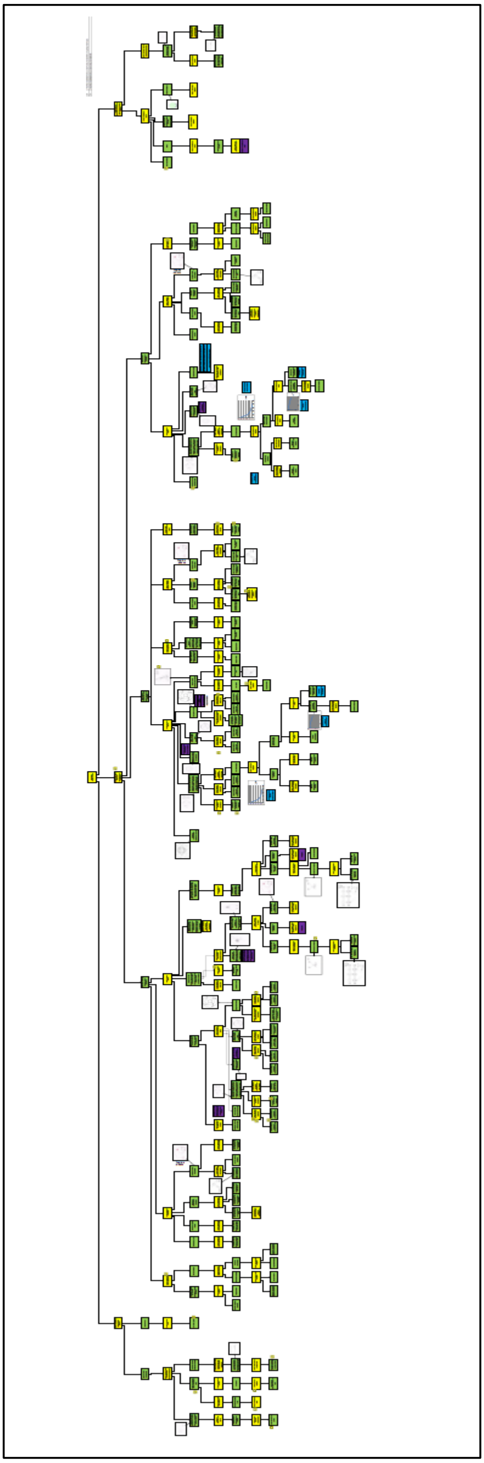


Figure 58. Overall view of the obtained PSN for the core cutter case study.

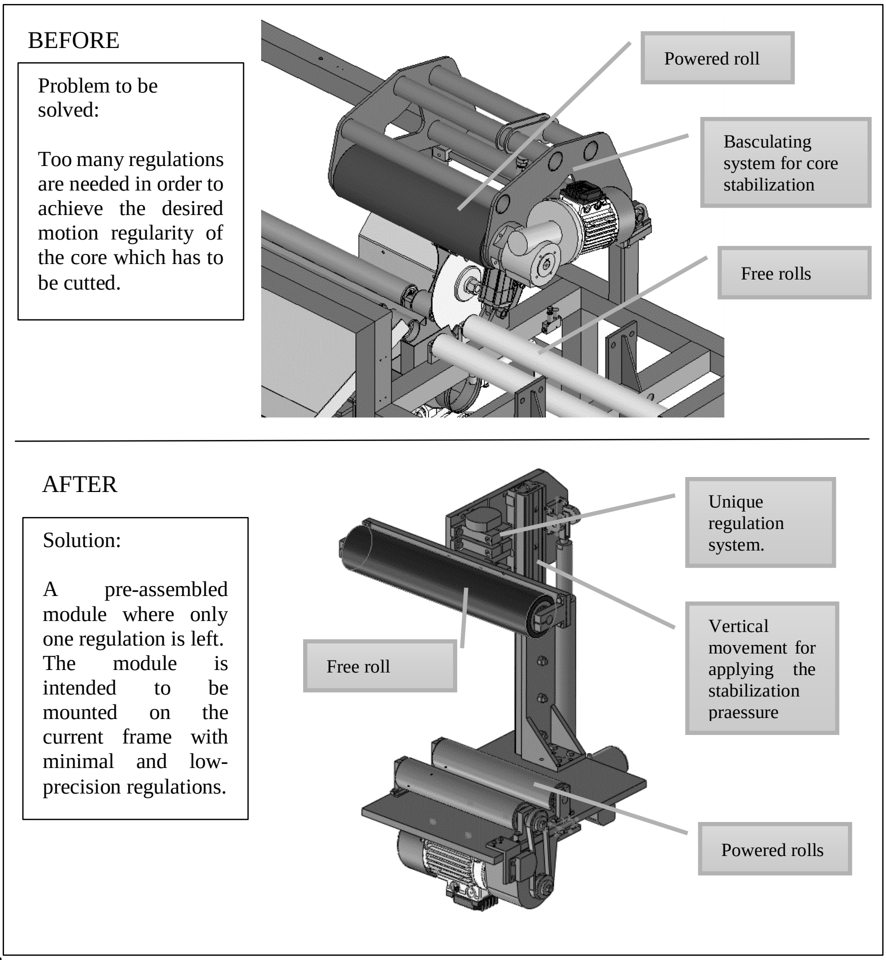


Figure 59. Cutting zone of the core cutter machine, before and after the complete re-design.

7.3.3.Considerations about the PSN approach for conceptual design

As reported in Section 2, functional decomposition and morphology approaches are characterized by some lacks recently highlighted by literature. In this sub-section discussions about the obtained results are presented according to the main lacks mentioned above. Furthermore, weak points of PSN and possible future developments are also discussed.

The first lack in the list reported at the end of Section 2 concerns the practical impossibility to avoid prejudices if a pure functional decomposition approach is pursued. As demonstrated in this section, thanks to the use of PSN it is possible to realize and visualize different problem decompositions at the same time, allowing also evaluating solution variants at different levels of abstraction and detail. In this way, thanks to the visualization of the relationships between problems and solutions, and also thanks to the set of rules for composing the PSN, any prejudice can be potentially identified by observing the net. Indeed, if the designer considers only a certain type of solution the network ramification will be very limited.

As stated above, the variety of problem decompositions and the related solutions are visualized at the same time, which implies that only a single schematization has to be realized. Moreover, the composition approach introduced in Section 6, allows using the same PSN in order to obtain different concept variants. Such characteristics of the proposed approach are considered here as a good remedy for the second lack listed at the end of Section 2, i.e. the noticeable effort of designers in realizing and managing many different functional models and related morphological charts.

For what concerns information gathering, the ASE process introduced in Section 6 allows to consider all the possible needs of further information during the solving process of a single design problem. Furthermore, the specific boxes of the PSN rapidly visualize where the information gathering activities are distributed into the conceptual design process, allowing the designer or the design team to manage them.

Eventually, the hierarchical distribution of problem and solutions characterizing the PSN, together with the set of rule and the ASE logic introduced in Section 6, allows mapping the conceptual design process in terms of how solutions and problems are interrelated, or in other words, how they co-evolve. Such a kind of peculiarity provides a rapid evaluation of how the design space has been explored, both in terms of quality and quantity. More precisely, it is possible to assess how many solution variants have been considered at each level of abstraction, and furthermore it is possible to evaluate how the designer has performed abstraction.

But actually, what is still missing? What are the lacks of PSN approach?

For sure, the first evidence that appears is the possibility to obtain very large networks in case of complex design tasks. By using commercial graphical software, it means that the PSN application process could be quite tedious and hard-working. However, what has been presented here must be considered as a first implementation of the approach, and should be intended to form the basis for the development of a future "tool", more user friendly for designers. Indeed, the algorithmic form of the process allows its implementation in a software with specific graphical characteristics, more manageable and intuitive for designers.

But before undertaking the development of a software more information are needed, concerning the applicability of the method, its easiness of use and in particular concerning the management of the abstraction levels.

Moreover, as observed in Section 6, some compatibility problems may arise during the combination of solutions belonging to different branches. The designer activity needed to solve such a kind of problems is not currently supported by the proposal, and then future research would be focused on that issue.

Those reported above and other research hints related to the possibility of using PSN and some main features, are currently considered by authors for their future research works, aimed at improving the efficacy of the conceptual design process.

7.4. Testing the PSN approach equipped with modularity tools.

The test described in this sub-section aims at evaluating the applicability of the PSN approach, integrated with the non-structured modularity tool described in Section 6. More specifically the objective of the present test can be split in two parts, i.e. it aims at verifying if the modularity approach implemented in the PSN actually works, and also at investigating the impact that the proposed conceptual design approach has on the quality of the solutions. Moreover, the application of the proposed method is supported by a first version of an interactive guideline, and observations about its ease of use and the impact on the correctness of the process applications are intended to be performed.

It has been performed on the same case study used for the test described in Sub-Section 6.2, i.e. the multifunction and customizable pen, by considering a new sample of engineering students. However, the level of expertise of the sample used here is quite different from that used in the previous test (see Sub-Section 6.2), since, in this case, the sample is composed by engineering students attending the course "Sviluppo e Ingegnerizzazione del Prodotto" of University of Florence where design methods and tools for product ideation and development are taught.

As for the other test, two groups have been considered, i.e. the Analysis (AG) and the Control (CG) ones. But in this case, student have more than five hours to complete the design task. Furthermore, both the groups were asked to realize three different concepts for the same requirement list, expressed as shown below:

- Allow the use of the pen, the pencil and the eraser in the same product.
- Allow to customize the product even after the purchase.
- It is required to allow the use of the eraser and the pencil concurrently.
- The product has to be easy to use, as for standard pens.
- The allowed product cost is "medium-high", taking as a reference the cost of the current variety of pens.

The CG was asked to perform the design task without the use of PSN. Indeed, they were not equipped with the necessary software and guidelines.

Instead, the AG was asked to use PSN, and they were supported by means of a software for constructing the problem-solution network (yEd software), and an interactive guideline (Appendix B). Such a guideline has been developed to assist the designer in the application of the rules and the tools described in Section 5.

7.4.1.Evaluation of the concepts

For the evaluation of the concepts produced by students, the same metrics introduced in Paragraph 7.2.2 have been used:

- Degree of fulfilment of the two main requirements. In order to assess how much the concept matches the design task.
- Degrees of Feasibility and usability of the solution. In order to purge the sample from unfeasible and/or unusable solutions.

- Modularity level. In order to assess modularity of the concepts, and to evaluate the effect of the proposal.

However, in order to evaluate the effect of PSN on the quality of results, another dimension has been investigated. Indeed, since PSN mainly aims at overcoming problems related to bias and preconceptions of the designer, the “novelty” level of the concepts has been measured. For that purpose, the method for assessing novelty proposed by Sarkar and Chakrabarti (2011) has been considered (Figure 61).

Such method is based on the model proposed by Srinivasan and Chakrabarti (2009), constituted by the construct listed below (from Sarkar and Chakrabarti 2011):

1. Phenomenon: interaction between system and its environment.
2. State change: change in property of the system (and environment) that is involved in interaction.
3. Effect: principle that governs interaction.
4. Action: abstract description or high-level interpretation of interaction
5. Input: physical quantity (material, energy or information) that comes from outside the system boundary, and is essential for interaction.
6. Organs: properties and conditions of system and environment required for interaction.
7. Parts: physical elements and interfaces that constitute system and environment.

The highest level of novelty, as shown in Figure 61, has been not considered in the test because it implies the definition of a new function for the product, not present in the State of the Art (SoA). But the design task and the related requirements have not been created in that sense, and students were explicitly asked to respect them.

Then, only three levels of novelty have been considered according to the selected metric:

1. Low: The concept is different from the SoA, only in parts or organs.
2. Medium: The concept is different from the SoA, in terms of parts or organs, and in terms of physical effects and physical phenomenon.
3. High: The concept is different from the SoA, in terms of parts or organs, in terms of physical effects and physical phenomenon, and in terms of inputs and state change.

However, it is worth to notice that it is impossible to consider the entire SoA for assessing novelty, because students may know only a part of it. Then the necessity to investigate how much the sample is acknowledged about the SoA arises.

For that purpose, a simple questionnaire has been dispensed to students fifteen days before the execution of the test. Such questionnaire, reported in Appenix C, shows many types of pens which can be observed by formulating research queries on most common Internet search engines. Moreover, the possibility to describe three additional pen types, not listed in the questionnaire, was left to students. However, no additional pen types was added and, moreover, only a limited number of students declared to be aware of all the pen types listed on the questionnaire. On the base of such observation, it is possible to claim that the considered sample of pen for the questionnaire was sufficiently exhaustive.

Then, once concepts have been produced by students, the novelty level has been assessed by considering a “personal SoA” for each student (Table 14).

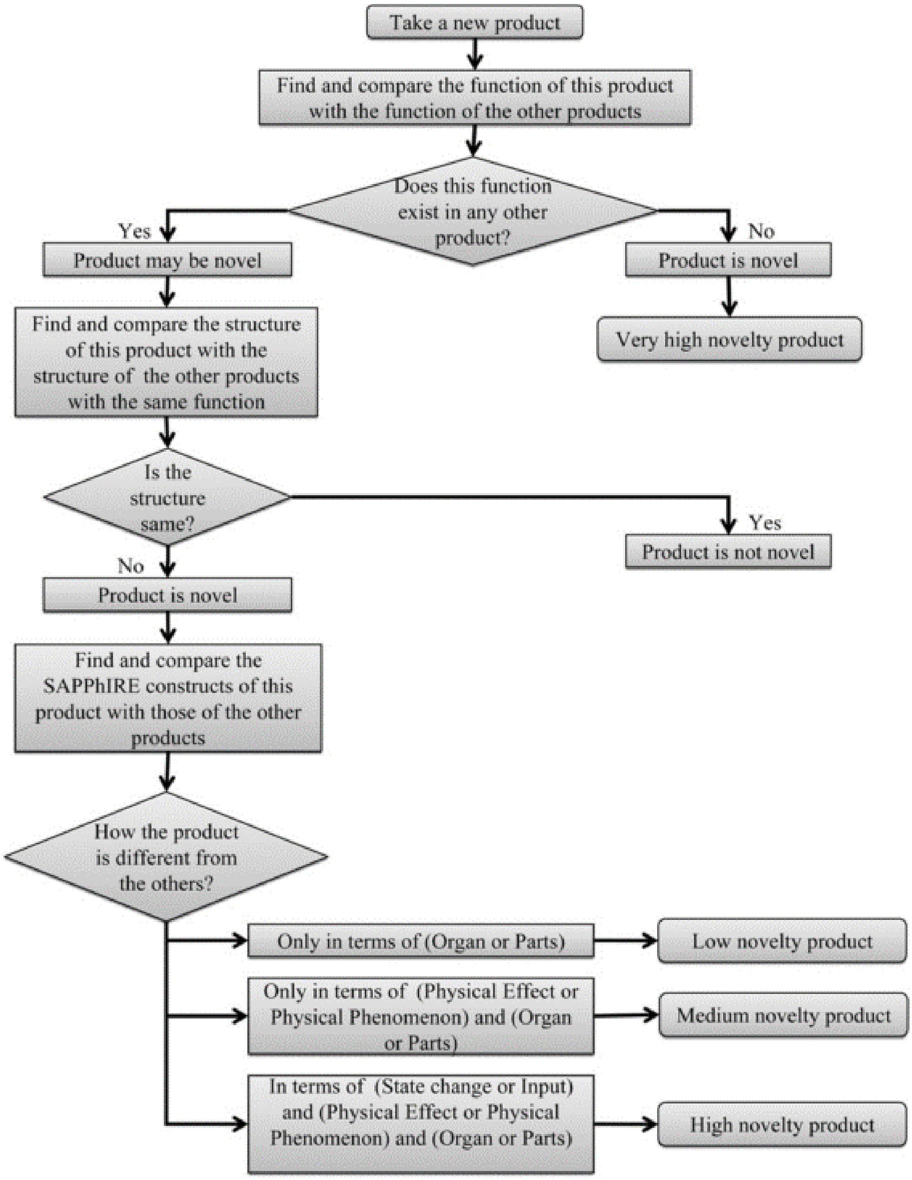
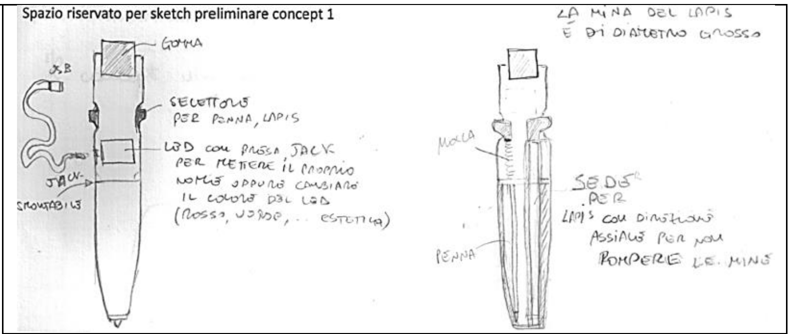
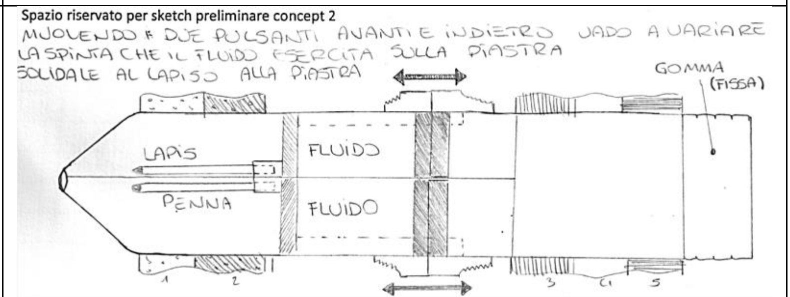
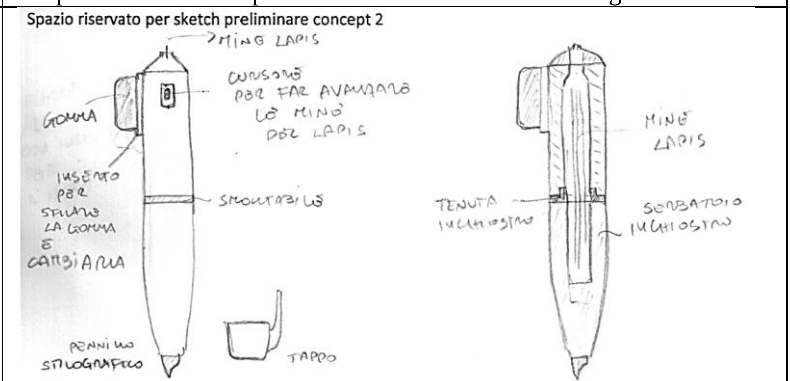


Figure 60. Method to assess the novelty of a product (from Sarkar and Chakrabarti, 2011).

Table 14. Examples of concepts at different levels of novelty.

<table>
<tr><td rowspan="2">High level of novelty</td><td>
</td></tr>
<tr><td><u>Short description:</u>
the pen has a led screen capable to be customized by means of a USB connection.</td></tr>
<tr><td rowspan="2">Medium level of novelty</td><td>
</td></tr>
<tr><td><u>Short description:</u>
the pen uses an incompressible fluid to select the writing means.</td></tr>
<tr><td rowspan="2">Low level of novelty</td><td>
</td></tr>
<tr><td><u>Short description:</u>
the pen is different to the SoA only in the arrangement of parts.</td></tr>
</table>

7.4.2.Results

Considering only a sample purged from unfeasible solutions (i.e. seven concepts were discarded), some not negligible evidences emerges, concerning the impact of PSN on results. Indeed, concerning the novelty level, it is possible to observe that the AG has a mean value sensibly higher than the CG (Figure 62).

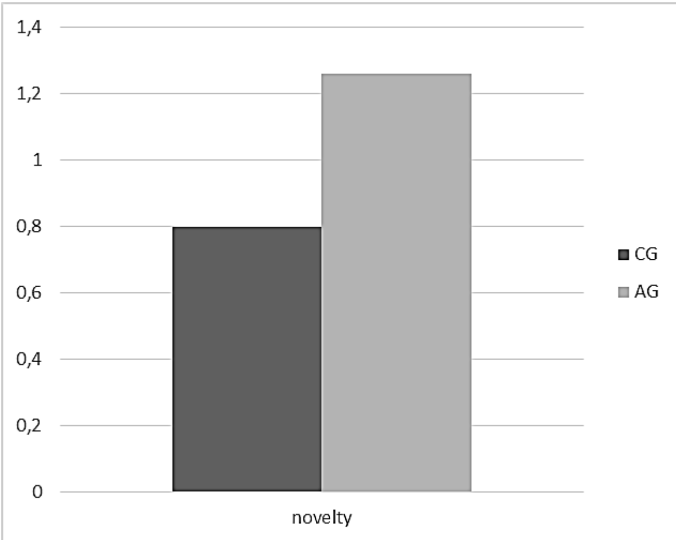


Figure 61. Mean values of the novelty levels for AG and CG.

The statistical reliability of the difference between the two sample can be demonstrated by a Two-sample t-test (Sheskin 2003), as shown in Figure 63. Such a test assert that the mean value of the novelty level for the AG is higher than the CG one, with 1.6 percent of uncertainty.

It means that the PSN gives an effective contribution in overcoming designer preconceptions, and then limiting the effect of psychological inertia.

Two-Sample T-Test and CI: Novelty; Group

```
Two-sample T for Novelty

Group   N   Mean  StDev  SE Mean
AG     27  1,259  0,859     0,17
CG     35  0,800  0,759     0,13


Difference = mu (AG) - mu (CG)
Estimate for difference:  0,459
95% lower bound for difference:  0,109
T-Test of difference = 0 (vs >): T-Value = 2,19  P-Value = 0,016  DF = 52
```

Figure 62. Results of the two-sample t-test executed with the Minitab software.

Figure 64 reports the differences observed on the mean values calculated for modularity levels of the concepts produced by students. It is possible to observe that the AG produced concepts with higher levels of modularity.

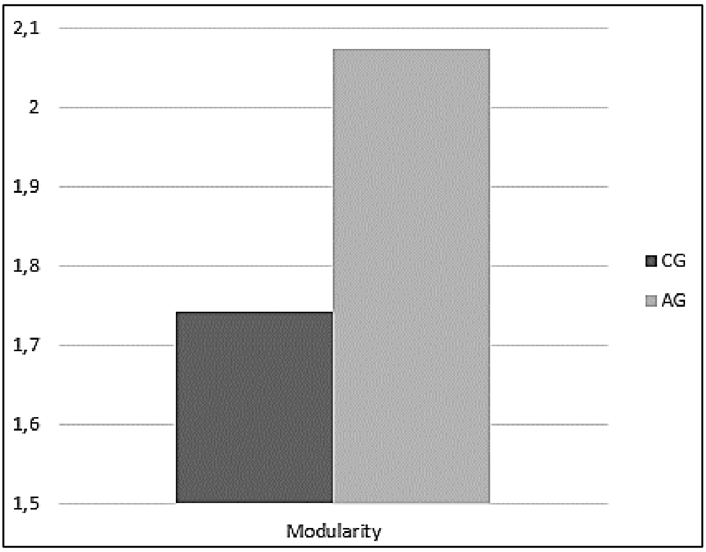


Figure 63. Mean values of the modularity level characterizing the concepts realized by the two groups.

However such an evaluation is not sufficient to assert that the proposal works well, because the mere maximization of modularity is not the objective of the present research work. Instead, it is necessary to investigate on the relationship between the satisfaction of the two main requirements and the application of the new approach.

Therefore, the mean values of the satisfaction levels of the main requirements have been calculated for both the groups, and reported on Figure 65.

Observing Figure 65 it is possible to note that AG and CG produces almost the same level of satisfaction for the first requirement. Therefore, one can infer that the use of the modularity tool of the PSN do not provide any contribution. However, by observing Figure 66, it is possible to note that for the specific case study (which is characterized by its specific set of objectives and constraints) there is a dependency between modularity level and the multifunction requirement, but it is not regular and almost negative. Then it is possible to assert that in this case, and considering the system level of detail, modular solutions are generally inferior to integral ones. This could justify the absence of any significant difference between AG and CG for the satisfaction level of the specific requirement, but it is hard to express a comprehensive evaluation operating on mean values. As a matter of the fact, AG produces solutions characterized by an higher mean value of the modularity level characterizing the produced concepts (Figure 64). This may implies that AG students concentrate their efforts in obtaining modularity for other purposes, and not for the fulfillment of the multifunction requirement, but it is impossible to demonstrate here such a statement.

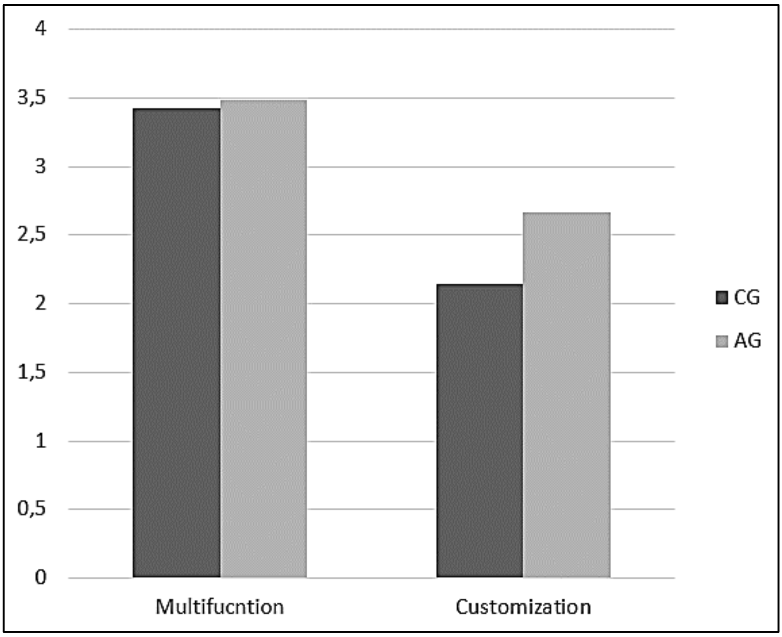


Figure 64. Mean values for the multifunction and customization satisfaction levels.

Instead, for what concerns the second requirement it is possible to observe a not negligible difference between the two groups (Figure 65). Such a difference can be statistically proved by the two-sample t-test shown in Figure 67, which reports that the mean value of the AG is superior to that of CG with an uncertainty of 2.4 percent. But in this case, as shown in Figure 68, there is a strong dependency between the satisfaction level of the requirement and the modularity level. Such a dependency is sensibly positive, and its existence, but not its trend, can be proved by the Chi-squared test (Sheskin 2003), reported in Figure 69.

Therefore, it is possible to suppose that the presence of modularity tools in the PSN approach, leads the AG to better results in terms of customization because an actual need of using modularity to solve the related problems has been observed on both the groups. In any case it is not possible to directly link the higher modularity value of AG to the better results reached for customization, because it is in contrast to what happened for the multifunction requirement. In other words, one can infer that AG focused its effort in developing modular solution to better fulfil the second requirement, however there is not a scientifically valid approach to demonstrate it with the available data.

In any case, it is possible to claim that where the need of modularity can be observed, the use of the PSN approach with the modularity add-on, led students to reach better results.

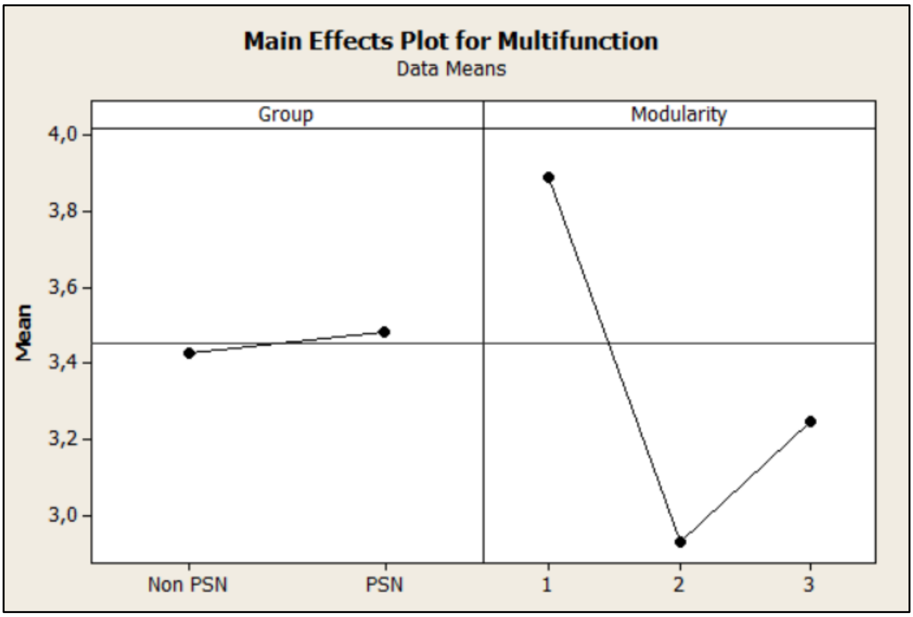


Figure 65. Mean values of the multifunction requirement satisfaction level, calculated for single groups (left) and for single modularity levels (right).

Two-Sample T-Test and CI: Customization; Group

```
Two-sample T for Customization

Group   N   Mean  StDev  SE Mean
AG     27   2,67   1,07     0,21
CG     35  2,143  0,912     0,15


Difference = mu (AG) - mu (CG)
Estimate for difference:  0,524
95% lower bound for difference:  0,092
T-Test of difference = 0 (vs >): T-Value = 2,03  P-Value = 0,024  DF = 50
```

Figure 66. Two-sample t-test performed to validate the difference between AG and CG for the satisfaction level of the customization requirement.

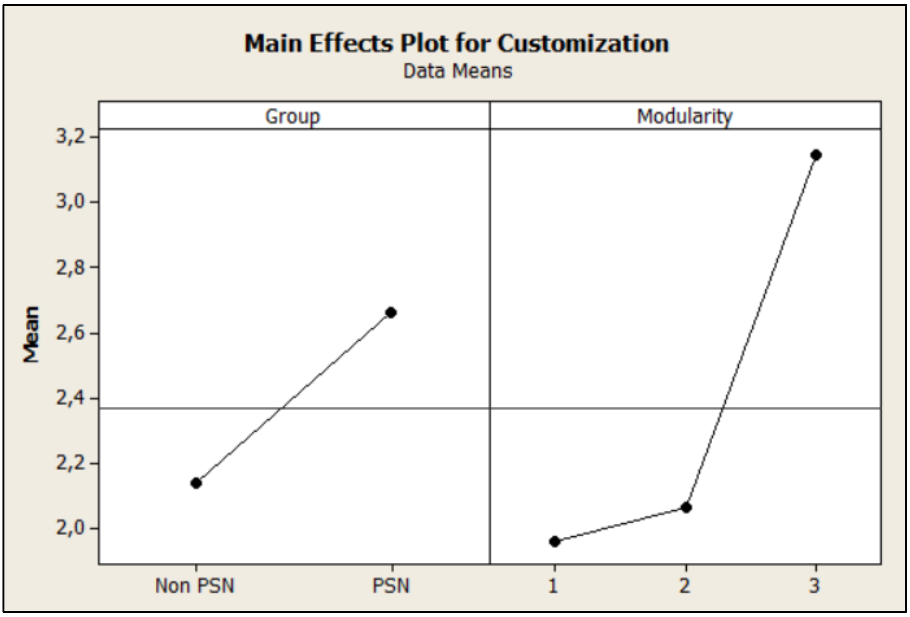


Figure 67. Mean values of the satisfaction level of the customization requirement, calculated for single groups (left) and for single modularity levels (right).

	Modularity levels						
	1		2		3		
Cust. Lev.	Observed	Expected	Observed	Expected	Observed	Expected	TOT
4	5	4,79032	1	2,66129	5	3,54839	11
3	0	6,09677	1	3,3871	13	4,51613	14
2	11	10,4516	11	5,80645	2	7,74194	24
1	11	5,66129	2	3,14516	0	4,19355	13
TOT	27		15		20		62
DoF:	6						
Critical CHI^2 5%:	12,592						
Calculated CHI^2:	43,9345	>di 12,592. Confirmed dependency					

Figure 68. Chi-squared test for verifying the existence of a dependency between the customization requirement satisfaction and modularity, for the present case study.

7.4.3.Considerations about results.

The results obtained in this test can be subdivided into two main groups:

- Validation of the positive effect of the PSN approach
- Validation of the proposed method for managing modularity during the conceptual design phase, implemented on a structured approach.

For what concerns the first point, the performed test demonstrates that the use of the PSN effectively leads the students toward a more comprehensive exploration of the design space. Indeed it has been shown that the novelty level of the concepts produced by students is higher for the group which has followed the PSN approach.

The modularity add-on, applied on the PSN (see 6.3), leads AG students to achieve an higher mean value for the related metric. A direct relationship between such level and the satisfaction of the two considered requirements is hard to be demonstrated here, however it has been shown that a not negligible dependency can be observed. Indeed, it has been shown that where it is possible to observe a generally valid need for modular solutions, AG reached better results.

However, what has been found with this test must be considered limited to the specific case study and dependent on the considered sample of convenience. A comprehensive test involving industrial engineers should be performed in order to obtain a sufficient reliability, but it has not been possible during the duration of this PhD.

Eventually, it has been noticed that the use of the interactive guideline led students to sensibly reduce doubts and errors in realizing the network, if compared with preliminary tests introduced in Section 6, concerning first PSN applications. However, a final and easy to use version of the toolset characterizing the PSN approach seems still far to be reached. Indeed, the use of a stand-alone software for the construction of the net, a stand-alone guideline not directly linked to the net and the impossibility to insert sketches on the net, make the application of the proposal quite onerous.

References

Altshuller, G. S. 1984. Creativity as an exact science: The Theory of the Solution of Inventive Problems. Gordon and Breach Science Publishers.

Bonnema, G. M., and Van Houten, F. J. A. M. 2006. "Use of models in conceptual design." Journal of Engineering Design 17 (6). 549–562. doi:10.1080/09544820600664994.

Boscaleri, A., Castagnoli, F., Rissone, P., & Corti, M., "Stratobus: A Multiuser Platform System for Making access to LDB Flight Easier and Cheaper", Proceedings of 19th Symposium on European Rocket and Balloon Programmes and Relates Research, ESA-SP-671, 2009, pp. 209-213.

Cascini, G., Rissone, P., Rotini, F., "Business re-engineering through the integration of methods and tools for process innovation", Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, Vol. 222:12, 2008, pp. 1715-1728

Collado-ruiz, D., & Capuz-rizo, S. F. (2013). Modularity and Ease of Disassembly: Study of Electrical and Electronic Equipment. Journal of Mechanical Design, 132(January 2010), 1–4. doi:10.1115/1.4000641.

Hölttä-Ott, K., De Weck, O., "Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints," Concurr. Eng., vol. 15, no. 2, pp. 113–126, 2007.

Hölttä-otto, K., Chiriac, N. A., Lysy, D., & Suk, E. (2012). Comparative analysis of coupling modularity metrics. Journal of Engineering Design, 23(November), 790–806.

Mattson, C.A. and Magleby, S.P., 2001. The influence of product modularity during concept selection of con- sumer products. ASME design engineering technical conferences, DETC2001. Paper No. DETC2001/DTM-21712, 9–12 September, Pittsburgh, PA.

Pahl, G., Beitz, W.," Engineering Design", 3rd ed., Springer-Verlag, London, 2007.

Pugh, S. 1991. Total Design Integrated Methods for Successful Product Engineering. Reading, Massachusetts: Addison-Wesley Publishing Company.

Salvador, F., "Toward a Product System Modularity Construct : Literature Review and Reconceptualization", IEEE Trans. Eng. Manag., vol. 54, no. 2, 2007, pp. 219–240

Sarkar, P., & Chakrabarti, A. (2011). Assessing design creativity. Design Studies, 32(4), 348–383. doi:10.1016/j.destud.2011.01.002

Sheskin, David J. Handbook of parametric and nonparametric statistical procedures. crc Press, 2003.

Srinivasan, V., & Chakrabarti, A. (2009). SAPPhIRE e An approach to analysis and synthesis, 17th International Conference on Engineering Design (ICED09), Stanford, USA, 23–27 August, 2009.

Ullman, D. G. 2010. The Mechanical Design Process. 4th ed. New York: Mc Graw Hill.

Ulrich K. (1995) The role of product architecture in the manufacturing firm. Research Policy, Vol. 24, no. 3, pp. 419-440.

Ulrich, K.T., Eppinger, S.D., “ Product Design and Development”, 4th ed. McGraw-Hill, Irwin, 2007

yED Software. http://www.yworks.com/en/products/yfiles/yed/

8. Discussions and future developments

As introduced in Section 1, the main objective of this research activity was to develop a proposal for facing modularity issues since early conceptual design activities. More in particular, seven "ideal" points have been indicated in Section 5, as specific goals to be reached by the proposal:

1) Identify early in the conceptual design process, the needs to apply modularity.
2) Guide the designer in generating correct modular solutions "before" knowing the overall solution, neither in terms of functions.
3) Avoid to move toward a maximization of modularity, but allow to reach the optimal product architecture, in terms of satisfaction of the requirement list characterizing the specific design task.
4) Reduce/avoid prejudices in decomposing a design problem.
5) Allow to easily generate and evaluate different possible concept variants characterized by different functionalities and different working principles.
6) Allow keeping track of the information gathering activities involved in the decisions taken during the conceptual design process.
7) Visualize the design space exploration.

Here in the following a discussion on the fulfilment of the above mentioned objectives is reported, also introducing what is still missing, together with possible future developments.

8.1. Identify the need to apply modularity

The investigation performed on the rise of modularity introduced in Section 6 and described in Section 7, allowed the author to speculate about a new way for identifying modularity needs without the necessity of any pre-existent physical or functional structure. A preliminary approach for non structured conceptual design approaches has been developed and the first test described in 7.2 showed that it can be used for such purposes. Moreover, the results of the last test described in this thesis confirm that the proposed approach can be implemented in a structured conceptual design method. Indeed, thanks to the use of modularity benefits, it has been possible to identify potential modularity needs by taking into consideration only the requirement list. As described in Section 6, the reliability of such identification process is based on the assumption that the benefits are generally valid, since extracted from years of literature contributions.

Nevertheless, the last performed test also highlights that the modularity benefits used for the identification of the opportunities for modular solutions, may lead to false indications. Indeed, it seems that even if considering similar case studies, the results of the identification is not stable and is dependent on the designer's expertise level. Indeed, taking as a reference the test described in 7.4, even a negative relationship can be found between modularity and the satisfaction level of the multifunction requirement.

However it is possible to assert that it is a secondary problem, because the proposed method guides the designer only in "evaluating" the possibility of consider modularity, not forcing him/her in adopting the modular solutions in the subsequent concept combination phase. Indeed, modularity benefits are widely acknowledged by literature and must be considered as generally valid, but each specific case study has to be cautiously examined by the designer.

Then, it would be interesting to evaluate all the benefits on a comprehensive set of industrial case studies, in order to find out when and why the positive relationship with modularity is confirmed and, conversely, which are the reasons that sometimes lead even to detrimental effects.

The possibility to develop a more accurate approach for the identification of the modularity needs may be an interesting new research activity, where the investigation approach proposed in 7.1 can be used as a valid tool. But it is worth to notice that it means to analyze a lot of industrial case studies and application in order to reach some significant information.

Moreover, it has to be taken into account also the fact that when facing a design task, sometimes it can be not trivial to assert if a compatibility between requirements and benefits actually exists. Then more efforts have to be spent in order to formulate more comprehensive definitions of the modularity benefits, maybe also by supporting the designer with an extended set of practical examples.

8.2. Guiding the designer in generating correct modular solutions.

The proposed approach for generating modular solutions, based on standard modularity types and implemented in the PSN approach, has been successfully tested as described in the previous section. Indeed, both in Sub-Sections 7.2 and 7.4, the AG was able to identify modular problems and to obtain modular solutions. As a matter of the fact, an higher mean value of modularity has been observed for such groups.

However it is possible to assert that such a solution generation process is heavily influenced on how the designer really understand the definitions of the various modularity types. Moreover, a major support has to be provided in order to guide the designer in selecting the level of detail or granularity (Chiriac et al. 2011) in which modularity has to be applied.

Similarly to what stated for modularity benefits, even in this case an extended set of practical examples may be a good way to improve the understanding of the various modularity types. However, future efforts should be focused also on defining a systematic support in the definition and the identification of the granularity level on which modularity has to be applied.

Despite the above mentioned lacks, anyway the proposed approach is actually extremely different from current modularity methods, and differently from them can be applied very early in the conceptual design process, even before knowing anything about the overall solution. Moreover, the explicit request to develop all of the three main characteristics (Interaction, Interface and Supply type), gives a not negligible help in conceiving a correct modular architecture for the solution.

8.3. Avoid to move towards a mere maximization of modularity.

As stated in Section 3, modularity is characterized also by detrimental effects and then the optimal modularity level has to be reached. Indeed a mere maximization of modularity may leads to a disadvantageous benefit-cost rate. Current modularity methods take somehow under control the above mentioned observation, but especially for the DSM approach, it can be hard to control the results since they are strongly dependent on the adopted clustering algorithm.

The proposal presented in this thesis does not aim at facing the problem of the optimal level of modularity, indeed no tools or rules have been provided here for that purpose. However this is not an inadvertence or a missing activity, but it has been a pondered decision. In fact, instead of measuring or assessing modularity levels and argue about them in relation to the requirement satisfaction, modular solutions are intended only as “solution variants” to be considered during the concept generation. In this way the concept variants are combined by selecting both modular and non-modular partial solutions from the PSN net. Then different concept variants can be obtained with different modularity rates. However the concept selection phase doesn’t take into account modularity, but only the evaluation parameters formulated for the specific task (i.e. how much the requirements have been fulfilled). In this way, the optimal architecture is implicitly selected.

Differently, it would be hard to identify the optimal level of modularity during the concept selection phase. More precisely, once the structure and the physical principles are selected, an optimization of the architecture is certainly possible. But since the conceptual design phase is characterized also by the choice of different working principles, it is very hard, or even not possible, to define a “generally valid” modularity value.

Then, on the base of the considerations reported above, it is possible to assert that the proposal completely match the present objective.

8.4. Reduction of prejudices in problem decomposition

The adoption of the PSN rules, together with the recursive application of the ASE logic and the formulation of the problems at different levels of abstraction and detail, seems to be a valid option in order to reduce prejudices when facing conceptual design tasks. Here in the following the reasons of such a statement are described. However it is worth of notice that the author of this thesis is conscious that the proposal presented here is very far to be considered a completely “objective” approach. Indeed, it hasn’t been possible to extract a generally valid rule for problem formulation, capable of leading different designers towards a unique “universally correct” network for each design task.

To tell the truth, the author is quite convinced that subjectivity can be reduced but cannot be completely eliminated till the design process is performed by humans. The reason is quite simple, i.e. since the coevolution of problems and solutions is strictly dependent by the solutions that designers are able to conceive, is the skill, the intelligence and the creativity of the designer (or the design team) which determinate the final PSN (or, generally, the final result). Moreover, designing is an activity performed by human designers for human stakeholders, and then an hypothetical “automatic designer” should be capable also of interpreting stakeholder emotions (which are surely subjective and not constant) and should be capable of communicating with him in order to gather and furnish project information in a optimal way. Fortunately (or unfortunately, depending on the point of view of the reader) only human designers are capable of doing that, and moreover, it is very hard to identify a unique designer’s profile capable of satisfying all possible stakeholders under every point of view.

However, concerning the problem-solving part of designing, it is widely acknowledged that subjectivity, which often leads to psychological inertia, has to be reduced as much as possible because it heavily limits the designer capability of exploring the design space (intended here as formed by problem space and solution space). It is also well acknowledged that abstraction is the most powerful weapon against psychological inertia.

The proposed approach, thanks to the rule concerning the “correct sequence of abstraction levels” (Paragraph 6.2.4), forces the designer in formulating problems and generating solutions at the highest possible level of abstraction. Moreover, the PSN graph allows to visualize not only the solution from which the problem has been spurted, but all the precedent and successive ramification (in other words, the origin of the problem and the consequences of the adopted solutions).

Then, while functional decomposition and morphology approaches necessarily needs to make implicit assumptions, the PSN approach allows exploring even “a plethora of assumptions” at different levels of abstraction and detail, but each of them can be visualized in the PSN graph. Indeed each of these assumptions coincides to each of the green boxes in the PSN (i.e. solutions), which in fact leads to different branches, i.e. different concepts.

In this way the designer can face the task with a perspective wider than that allowed by the morphological charts, which are related only to specific functional models.

However, except for the comparison performed in Sub-Section 7.3, it has not been possible to perform tests focused on the evaluation of the actual reduction of prejudices when adopting PSN instead that functional decomposition and morphology. Future works on PSN should certainly be focused on such an investigation, in order to assess the validity of the proposal.

Nevertheless, a partial validation of the proposed approach are the results of the test described in 7.4, which show that for the considered case study the PSN approach actually brings to a deeper exploration of the design space. Indeed, students who applied the PSN obtained concepts characterized with an higher level of novelty. However, two important observations must be reported. The first is that the test has been performed for a simple and academic case study. It implies that many other tests has to be performed in order to comprehensively validate the efficacy of the proposal. The second observation concerns the sample of convenience considered for the test. Indeed, testing the proposal on engineering students may be a valid way to investigate preliminary characteristic of a methodological proposal, however the real impact on the industrial world is missing. But in order to allow the use of the approach directly by firm's engineers, there is the need to implement the logic into a software equipped with a simple and intuitive GUI. Indeed, the influence of the graphical interface used to guide the generation of the network has been observed also on students. However, the informatics tools currently used are extremely rudimental and need to be upgraded with more comprehensive and tailored informatics means.

8.5. Generation and evaluation of different concept variants

Thanks to the problem-solution structure of the PSN approach and its graphical tool it is possible to represent all the possible variants in a unique schematization (i.e. the PSN graph). Indeed, the PSN is a sort of morphological tool where all the different "ways" for composing the concept are represented.

Differently from morphological charts, PSN allows to visualize the entire considered solution space. Indeed morphological charts, in the Pahl and Beitz approach represent only the working principles for implementing functions of a specific functional structure. Instead, PSN shows functions, working principles and structures (when possible) together.

Then, during the proposed concept composition approach, it is possible to obtain concepts with completely different functional structures, without the necessity of working on other graphical tools. Finally, the obtained concepts (in form of sketches and textual and/or graphical descriptions) can be evaluated, e.g. with the well-known approaches introduced in Section 2.

Therefore, it is possible to assert that the proposed approach is capable of generating an undefined number of concept variants, which differ not only in the physical structure, but also in functions and working principles. Moreover, thanks to the adoption of the well-known selection processes introduced in Section 2, the proposed approach allows a comprehensive evaluation of the concepts in relation to the list of requirements.

8.6. Allow keeping track of the information gathering.

The ASE logic presented in this thesis has been conceived in order to model the necessary information gathering activities that occur during the problem-solving process, intended as a fundamental part of designing. It has been shown that the need of information may arise in each of the ASE steps, i.e. both concerning problems and solutions.

In the PSN, a special type of boxes had been created in order to visualize the need of information and the related documentation (see info boxes of Figure 29 where, for example if using MS Visio, it is possible to insert an hyperlink to an external document containing the gathered information).

Moreover, it is possible to manage the information gathering activities simply by marking the information boxes with generic symbols representing the state of the activity and its overall results, i.e. “work in progress”(Figure 70, a), “information acquired” (Figure 70, b) and “missing information” (Figure 70, c).

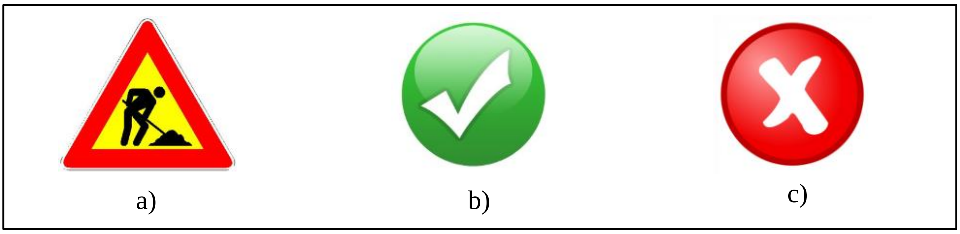


Figure 69. Generic example of possible symbols for indicating the state of the various information gathering activities visualized in the PSN.

Such a peculiarity is fundamental for the application of the “completeness rule” introduced in Sub-Section 6.2.

8.7. Visualize the design space exploration.

The PSN graph allows to visualize how the design space has been explored both in terms of extension and quality. In order to explain such a statement it is necessary to recall the Synthesis step of the ASE logic and the rule concerning the correct sequence of the abstraction levels (Section 6).

For those concerning “how much” the design space has been explored it is sufficient to observe the horizontal extension of the network, which represents how many ramification variants have been created during the concept generation phase. Indeed, the horizontal extension of the net mainly depends on how many different solutions have been taken into considerations for each problem. As a matter of the fact, it is sufficient to observe the generic example shown in Figure 71, where two different “types” of design space explorations have been represented for the same design task. In such an example it is possible to notice that considering only one solution for each problem leads to a very “thin” network (Figure 71). Conversely, if the designer considers more than one alternative for solving the problems (for the various levels of abstraction and detail), the network results sensibly “larger” (Figure 72). However, it does not imply that a mere maximization of the number of the solution variants is sufficient to demonstrate a better design space exploration. Indeed, each solution proposed in the synthesis step of the ASE must be correctly evaluated in the evaluation step, before being placed in the PSN graph.

The evaluation of the coherence of the solutions with the design task and the available resources is certainly a very important step to be done in order to assess the quality of the design process. However a more interesting evaluation can be performed by observing the PSN graph, i.e. that concerning “where” the design space exploration has been focused. More specifically, simply by observing the extension of branches and the related problem-solution sequences, it is possible to understand the “abstraction levels” of the solution variants. Indeed, it may be quite simple to generate many solutions at a low level of abstraction, e.g. different solutions sharing the same working principle. But a comprehensive conceptual design process should be pointed also at searching for radically different ways for fulfilling the list of requirements (obviously, accordingly to the available resources). Observing the distribution of solutions in the PSN it is possible to visualize if solutions have been searched also at high levels of abstraction.

Moreover, the PSN allows to show the distribution of the efforts on the various parts of the design task. Indeed, observing the extension of branches and the quantity of information gathering activities, it is possible to understand where efforts were focused during the design process (e.g. one of the main functional problems may results not sufficiently investigated due to the poor variety of solutions).

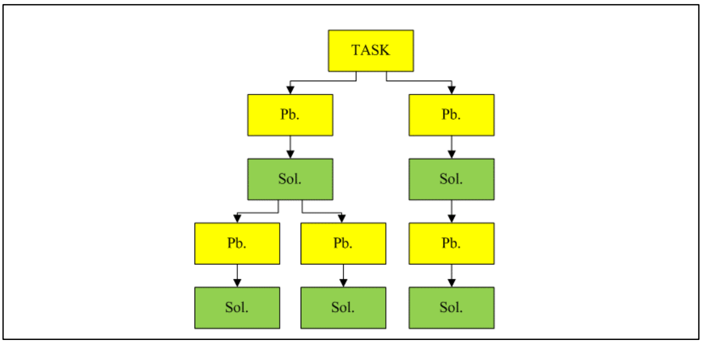


Figure 70. Generic example showing a poor design space exploration.

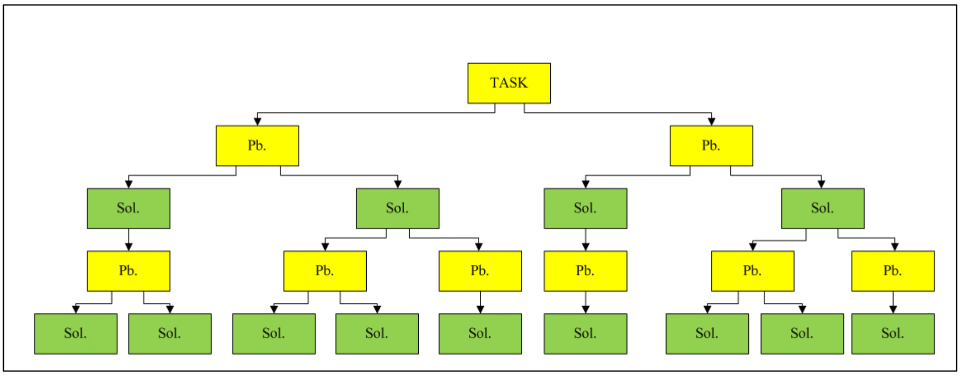


Figure 71. Generic example showing a more extended design space exploration, for the same design task of Figure 70.

8.8. Final considerations

The research activity discussed in this thesis led to the development of a new conceptual design approach capable of facing modularity issues without the need of generating preliminary concepts. The obtained results have been deeply discussed in this section, in relation to the ideal objectives introduced in Section 5. Moreover, it has been highlighted what is still missing, or only partially achieved, indicating possible future developments. More in particular, the following issues have to be faced in future activities:

- There is a need of further investigations on the link between modularity benefits and the adoption of modularity.

  It is well acknowledged in literature that the effect of modularity is far to be fully understood. However the approach proposed in 7.1 seems to be a valid tool for performing further investigations. As a matter of the fact, it has been positively judged by scientific community (see Appendix E).

- A comprehensive management of the abstraction level is still missing.

  In the current version it has not been possible to obtain a comprehensive and generally valid guideline for formulating problems and generating solutions at a highest level of abstraction. It implies that there is an actual need to give supplemental support to novices, especially on first uses of the new proposed approach. Then, future research activities should be focused on a better understanding of abstraction issues and at conceiving a more structured guideline for novices.

- The new approach for conceptual design has to be tested on more reliable samples, i.e. designers belonging to the industrial world.

  Indeed, beyond the extension of the tests on a more extended sample, it has to be evaluated how really the proposed approach can overcome the problematic related to the reluctance of designers in using systematic approaches. However, in order to perform such an evaluation, a tailored software tool has to be developed. Indeed, since the proposal is based on set of rules and a fundamental logic, the understanding of them implies an initial effort that can be certainly source of reluctance. Moreover, the construction of the network, if performed with standard commercial software, may results quite tedious. However, a tailored software equipped with an efficient “start-up guideline” completely interactive and linked to the generation of the PSN could be a powerful tool to reduce the initial impact of designer on this approach. That developed in this work (Appendix B), which has used in the last performed test, can be considered a very first (non-automatic) prototype of that guideline.

References

Chiriac N., Holtta-Otto K., Lysy D., Suk Suh (2011) E. Level of Modularity and Different Levels of System Granularity. Journal of Mechanical Design, vol. 133, no. 10.

Pahl, G., Beitz, W.," Engineering Design", 3rd ed., Springer-Verlag, London, 2007.

9. Conclusions

Starting from preliminary evaluations, the research activity described in this thesis was focused on the development of a proposal capable of facing modularity issues during early conceptual design activities.

A comprehensive literature analysis has been performed in order to validate the initial hypothesis and to acquire the needed knowledge base for facing the research issues. The results of such analysis showed the lacks of current modularity methods and, moreover, identified the limits imposed by one of the most acknowledged current conceptual design approaches. Therefore, the main objective was split into two parts, i.e. the development of a new way for achieving modularity and the development of a structured conceptual design approach capable to implement it.

Preliminary proposals have been produced and tested, concerning the identification of the need to apply modularity and the generation of modular solutions. Moreover, a new conceptual design approach has been conceived, taking inspiration from literature contributions, but adding some not negligible integrations. These results led to the so called PSN approach, equipped with specific modularity tools capable of operating without the need of any pre-determined concept.

In order to validate the proposal, some tests have been performed on samples of convenience composed by engineering students, bringing to first confirmations. More specifically, the tests showed that the PSN approach can give not negligible contributions in obtaining novel solutions, partially demonstrating the positive effect of the fundamental logic (ASE) developed in this research activity, which forms the core of the proposed conceptual design approach. Indeed, such a logic guides the designer in facing the problem-solving process and allows to comprehensively map the information gathering activities. It has been shown also that the proposed approach for facing modularity issues, integrated in the new conceptual design method, led to positive results. Indeed, students conceived concepts characterized by an higher level of modularity, whereas the design task was intentionally chosen from those potentially solvable with modularity.

Eventually, the results of the research activity have been deeply discussed, pondering on achieved objectives and highlighting what is still missing. On the base of such observations, some possible future activities have been inferred, concerning refinements and new developments of the proposal presented in this thesis.

Acknowledgements

The present thesis work was carried out in the Department of Industrial Engineering of the University of Florence (DIEF). Firstly, I would like to express my sincere gratitude to my supervisors Prof. Eng. Paolo Rissone and Eng. Federico Rotini PhD. Special thanks goes also to my colleagues Dr. Francesco Saverio Frillici, Eng. Yuri Borgianni and Eng. Daniele Bacciotti, for their participation on some of the activities involved in my research work.

I would like to sincerely and gratefully thank Prof. Eng. Roberto Viganò, for his important comments and suggestions, which contribute to reach my best for the editing of this thesis and to better focus future efforts.

Eventually, I want to express my sincere gratitude to all the members of the DIEF PhD commission, for their precious comments, suggestions and impressions reported during the three years of this scholarship, helping me in refining the objectives and achieving the results shown in this thesis.

Appendix A – Further definitions and classifications

A.1. Definitions about Design

Design Methodology

"Many authors have proposed theories, models and methods in their search to explain or improve upon aspects of design practice. This field of literature, commonly known as design methodology…" D. Wynn and J. Clarkson, "Chapter 1 Models of designing," in *Design Process Improvement - A Review of Current Practice.*, 2005, pp. 34–59.
"Design methodology is the science of methods that are or can be applied in designing. In English, the word 'methodology' has two meanings. The first meaning is: a science or study of method, i.e. the description, explanation and valuation of methods. The second meaning of 'methodology' is: a body of methods, procedures, working concepts and rules employed by a particular science, art or discipline." Roozenburg and Eekels, *Product_design_fondamental_etc .pdf*. John Wiley and sons, Inc, 1991.
"Perhaps, a classic view is that a design theory is about how to model and understand design, while design methodologies are about how to design or how design should be." Tomiyama, T., Gu, P., Jin, Y., Lutters, D., Kind, C., & Kimura, F. (2009). Design methodologies: Industrial and educational applications. CIRP Annals - Manufacturing Technology, 58(2), 543–565. doi:10.1016/j.cirp.2009.09.003

"Design theory is about design; it explains what design is or what is being done when designing. On the other hand, design methodology is a collection of procedures, tools and techniques for designers to use when designing. Design methodology is prescriptive as it indicates how to do design, while design theory is descriptive as it indicates what design is." N. F. O. Evbuomwan, S. Sivaloganathan, and A. Jebb, "A survey of design philosophies, models, methods and systems," J. Eng. Manuf., vol. 210, pp. 301–320, 1996.

Model and Method

"In a sense, any identifiable way of working, within the context of designing, can be considered to be a design method. .... Design methods can therefore be any procedures, techniques, aids or tool for designing" N. Cross, *Engineering design Methods_Strategies for Product Design_ Third Ed.* 2000.
1) *A method is a specific way to proceed* 2) *A method is a rational procedure* 3) *A method is general – that means: applicable to more than one problem* 4) *The use of a method is observable* Roozenburg and Eekels, *Product_design_fondamental_etc .pdf*. John Wiley and sons, Inc, 1991.
• *Models, which refer to a description or prescription of the morphological form of the design process.* • *Methods, which prescribe systematic procedures to support the stages within a model.* D. Wynn and J. Clarkson, "Chapter 1 Models of designing," in *Design Process Improvement - A Review of Current Practice.*, 2005, pp. 34–59.

Design

"People have always designed things. One of the most basic characteristics of human beings is that they make a wide range of tools and other artefacts to suit their own purposes. As those purposes change, and as people reflect on the currently-available artefacts, so refinements are made to the artefacts, and sometimes completely new kinds of artefacts are conceived and made. " N. Cross, *Engineering design Methods_Strategies for Product Design_ Third Ed.* 2000.

Design (continue)

"The process of establishing requirements based on human needs, transforming them into performance specification and functions, which are then mapped and converted (subject to constraints) into design solutions (using creativity, scientific principles and technical knowledge) that can be economically manufactured and produced."

N. F. O. Evbuomwan, S. Sivaloganathan, and A. Jebb, "A survey of design philosophies, models, methods and systems," J. Eng. Manuf., vol. 210, pp. 301–320, 1996.

Design Problems

- *The designer's difficulties are therefore two-fold: understanding the problem and finding a solution. Often these two complementary aspects of design (problem and solution) have to be developed side-by-side. The designer makes a solution proposal and uses that to help understand what the problem really is and what appropriate solutions might be like.*
- *The kinds of problem that designers tackle are regarded as ill-defined or ill-structured, in contrast to well-defined or well- structured problems such as chess-playing, crossword puzzles or standard calculations.*
- *There is no definitive formulation of the problem, Any problem formulation may embody inconsistencies, Formulations of the problem are solution-dependent. Proposing solutions is a means of understanding the problem, There is no definitive solution to the problem.*
- *In particular, sub-solutions can be found to be inter-connected with each other in ways that form a pernicious, circular structure to the problem, e.g. a sub-solution that resolves a particular sub-problem may create irreconcilable conflicts with other sub-problems.*

N. Cross, *Engineering design Methods_Strategies for Product Design_ Third Ed.* 2000.

A.2. Classifications of design models and methods

N. Cross, *Engineering design Methods_Strategies for Product Design_ Third Ed.* 2000.

Design Models	Descriptive Models (describe the sequences of activities that typically occur in designing)
	Prescriptive Models (attempt to prescribe a better or more appropriate pattern of activities)
Design methods	Creative methods
	Rational methods

N. F. O. Evbuomwan, S. Sivaloganathan, and A. Jebb, “A survey of design philosophies, models, methods and systems,” J. Eng. Manuf., vol. 210, pp. 301–320, 1996.

Design philosophies	Semantic school
	Syntax school
	Past experience school
Design models	Prescriptive based on the design process
	Prescriptive based on product attributes
	Descriptive models
Design methods	• Methods intended to provide basic improvements in the way designers work. • Methods that act on the creative characteristics of human being. • Methods that attempts to describe and master the problem situation by means of strict logic and mathematics. • Methods that prescribe methodical rules and regulations, which can significantly increase the overall probability of success. • Methods based particularly on the knowledge of the artifact being designed. • Methods that encourage the use of technical means and aids and aims at the automation of that part of the design process. • Combination of the above methods appropriate to the existing situation.

D. Wynn and J. Clarkson, “Chapter 1 Models of designing,” in *Design Process Improvement - A Review of Current Practice.*, 2005, pp. 34–59.

Design models (Design focused literature)	Interrelated dimension 1	Stage based	*…purely stage-based models indicate only the possibility of rework using feedback loops between stages*
		Activity based	*…the highly cyclical, rework-intensive activities characteristic of the designer's day-to-day activities as the problem-solving dimension*
	Interrelated dimension 2	Problem oriented	*in which the emphasis is placed upon abstraction and thorough analysis of the problem structure before generating a range of possible solutions.*
		Solution oriented	*in which an initial solution is proposed, analysed and then repeatedly modified as the design space and requirements are explored together*
	Interrelated dimension	Abstract	*which are proposed to describe the design process at a high level of abstraction. Such literature is often relevant to a broad range of situations, but does not offer specific guidance useful for process improvement.*
		Procedural	*which are more concrete in nature and focused on a specific aspect of the design project. They are less general than abstract approaches, but more relevant to practical situations.*
		Analytical	*which are used to describe particular instances of design projects. Such approaches consist of two parts: a representation used to describe aspects of a design project, such as the design structure matrix or DSM (Steward, 1981); and techniques, procedures or computer tools, which make use of the representation to understand better or improve the process of design;*

A.3. Other definitions of product architecture and modularity

Hölttä-Otto, K., “Modular Product Platform Design”, Doctoral Dissertation, Helsinki University of Technology. , 2005.

- *The US Department of Defense, on the other hand, use more life cycle thinking in their definition of architecture: The structure of components, their relationships and the principles and guidelines governing their design and evolution over time.*
- *Maier and Rechtin have a systems approach and include the process in their definition: The structure (in terms of components, connections, and constraints) of a product, process, or element.*
- *Crawley et al. give a similar definition for system architecture, but instead of physical components they refer to entities that could be functions, physical or nonphysical "components", etc.: System architecture is an abstract description of the entities of a system and the relationships between those entities.*
- *Hollta-Otto: System architecture is an abstract description of the entities of a system and the relationships between those entities and the scheme by which these entities are mapped to larger physical or non-physical sub-systems of a system.*
- *O'Grady defines "hard" and "soft" modules. "Hard" modules are physical assemblable modules and "soft" modules have limited physical presence e.g. software, service, financial products, insurance, etc*
- *Mattson and Magleby divide modularity into three categories: design,manufacturing, and customer modularity*
- *Gershenson categorizes modules into the design and manufacturing, as well as the end-of-life modularities.*
- *Hubka and Eder define a modular design as "connecting the constructional elements into suitable groups from which many variants of technical systems can be assembled".*
- *Salhieh and Kamrani define module as "building block that can be grouped with other building blocks to form a variety of products". They also add that modules perform discrete functions, and modular design emphasizes minimization of interactions between components.*
- *Otto and Wood: "product modules are defined as integral physical product substructures that have a one-to-one* correspondence with a subset of a product's functional model".
- *Ericsson and Erixon add that in addition to the similarity between the physical and functional architecture of a product, a module should have minimal interaction with other modules or the rest of the system*
- *Baldwin and Clark define a module as "a unit whose structural elements are powerfully connected among themselves and relatively weakly connected to elements in other units*

Pahl, G., Beitz, W.,” Engineering Design”, 3rd ed., Springer-Verlag, London, 2007.

Modularity is the degree of purposeful structuring of the product architecture.

Modularisation *is the purposeful structuring of a product in order to increase its modularity. The aim is to optimise an existing product architecture to meet product requirements or to rationalise production processes.*

Modules *are units that can be described functionally and physically and are essentially independent*

Kamrani, A.K. and Salhieh, S. M., 2002. Product Design for Modularity, 2nd ed. Kluwer Academic Publishers.

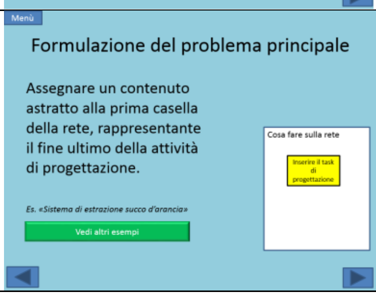
Modularity *aims to identify the independent, standardized, or interchangeable units to satisfy a variety of functions. Modularity can be applied in the areas of product design, design problems, production systems, or all three. It is preferable to use modular design in all three types at the same time*

Appendix B – Interactive guideline for applying the PSN approach.

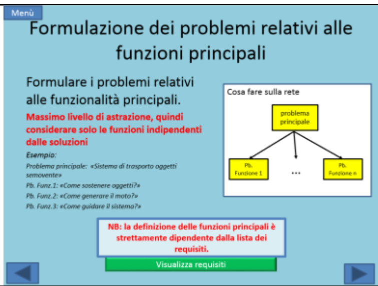
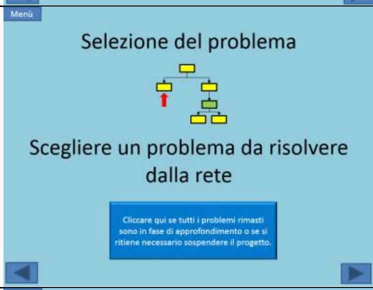
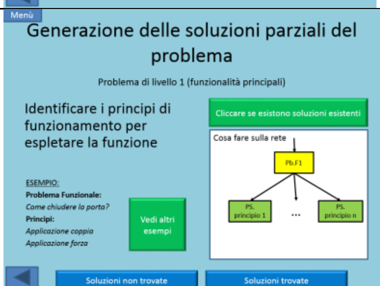
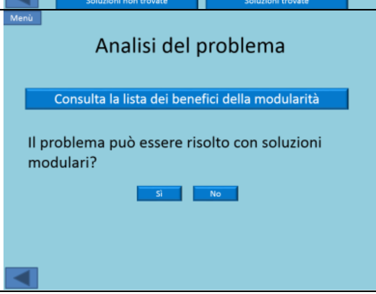
<table>
<tr><th>Slide</th><th>Short description</th></tr>
<tr><td>

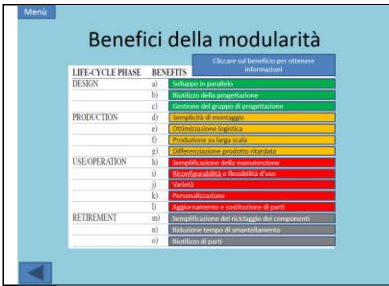

</td><td>Initial slide.
It is possible to start a new project or to continue an existing one, considering saved requirements.</td></tr>
<tr><td>


</td><td>Main menu.
It can be recalled from all of the slides composing the guideline.
All of the various steps can be reached by clicking the related active button.</td></tr>
<tr><td>


</td><td>Inserting objectives and constraints.
Here the designer is asked to split requirements into the types above.
It is impossible to proceed further in the guideline if at least one element for each column hasn't been inserted.</td></tr>
<tr><td>


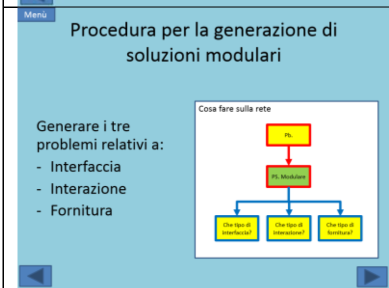
</td><td>Formulating the principal problem, i.e. the Design Task.</td></tr>
</table>

<table>
<tr>
<td>
</td>
<td>Formulating the first level of functional problems.</td>
</tr>
<tr>
<td>
</td>
<td>Selecting the problem to be solved from the network.</td>
</tr>
<tr>
<td>
</td>
<td>Selecting the level characterizing the selected problem.</td>
</tr>
<tr>
<td>
</td>
<td>Generating solution for problems belonging to the first level.</td>
</tr>
<tr>
<td>
</td>
<td>Generating solution for problems belonging to level higher than one.
Here the designer is asked to compare the problem with modularity benefits, in order to evaluate the possibility to develop modular solutions.</td>
</tr>
</table>

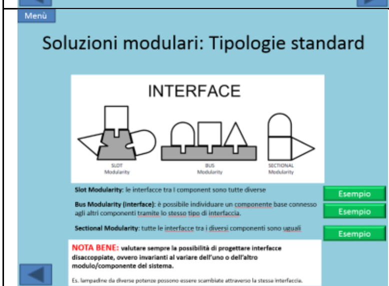


Modularity benefits.
Each of them is described by mean of examples, approachable by clicking on the specific box.

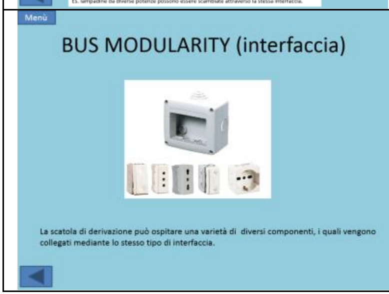
One of the examples used to describe the modularity benefits.



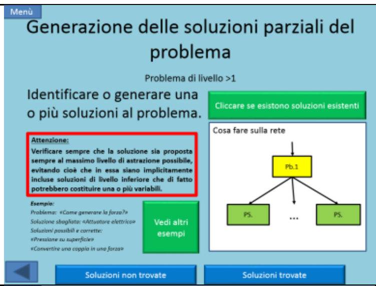
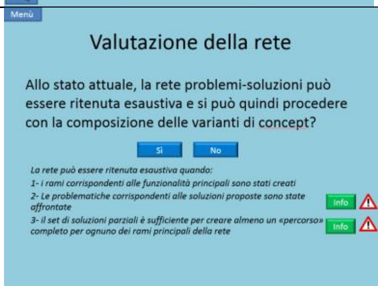
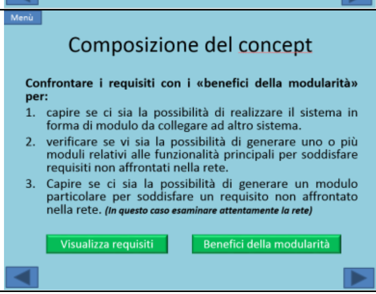
Procedure to be followed in case of modular solutions.
The designer is asked to conceive solutions taking inspiration from standard modularity types.



One of the slides representing the groups of standard modularity types.
Each type is explained by mean of examples.



One of the examples used to explain modularity types.

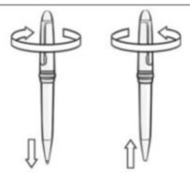
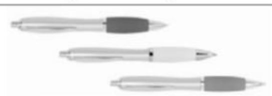
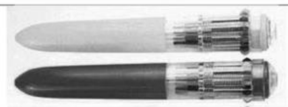
	Genereting solution for non-modular problems, i.e. problems which are not compatible with any of the possible modularity types.
	Evaluating the proposed solutions. Proposed solutions are evaluated, as foreseen in the ASE logic. Indeed, also the possibility of information gathering activities is considered, by clicking the lower active box.
	Evaluation of the PSN. As foreseen in the rules for the PSN approach, the completeness of the network has to be evaluated in order to decide what to do. Indeed, in case of incomplete network, the problem and solution generation has to be continued.
	Concept composition. Once a complete PSN has been reached, the next step is to compose the concept by choosing specific branches for each of the main functional problems.
	Evaluating modularity in concept composition. Requirements and modularity benefits are compared in order to identify needs for modular assemblies.

	Selecting the type of composition: System as a module to be connected to a supersystem. Modular system composed by some modules. Non-modular system.
	Instructions for generating a modular system.
	Instructions for sketching a modular system, composed by specific modules.
	Instructions for sketching a non-modular system.
	Concluding slide.

Appendix C – Questionnaire for investigating the background of the sample of convenience.

Segna le tipologie che hai utilizzato

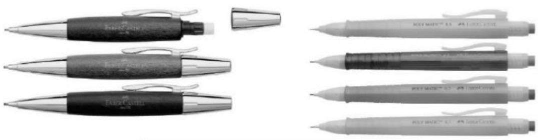
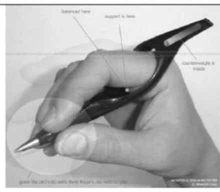
Segna le tipologie ti cui conosci i dettagli di struttura e funzionamento

Biro statica			
Stilografica			
Penna a punta sintetica			
Biro a scatto con pulsante			
Biro a scatto con pulsante laterale			
Biro a scatto twist			
Penna con impugnatura personalizzata			
Biro multicolore			

NOME________ COGNOME________

MATRICOLA________

Biro cancellabile			
Penne con intera struttura di forma personalizzata			
Penne a componenti impilabili			
Penne con struttura parzialmente personalizzabile			
Penne con grafica personalizzabile			
Lapis con involucro in legno (senza gomma)			
Lapis in legno + gomma			
Portamine con mina fine e gomma sotto il tappo			
Portamine per mine spesse e gomma sotto il tappo			

Portamine maxi (mina molto spessa)			
Portamine personalizzati			
Penne multifunzione a selezione			
Penna ad anello			
Penna con evidenziatore			
Penna con estremità Biro/Stilografica intercambiabile			

Altro tipo di penna, lapis o portamine conosciuti	
Tipologia	Descrizione/rappresentazione schematica

Appendix D – Publications (First pages)

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 00 (2014) 000–000

www.elsevier.com/locate/procedia

TRIZ FUTURE CONFERENCE 2014 - Global Innovation Convention

Linking TRIZ to Conceptual Design Engineering Approaches

Francesco Saverio Frillici[a,*], Lorenzo Fiorineschi[a], Gaetano Cascini[b]

[a]*University of Florence, via di Santa Marta, 50139, Florence, Italy*
[b]*Politecnico di Milano, via La Masa, 20156 Milan, Italy*

* Corresponding author. Tel.: +39-055-479-6514 ; fax: +39-055-479-6525. *E-mail address:* francescosaverio.frillici@unifi.it

Abstract

During the last decades, product design has yielded several interest by scholars, leading to a great amount of contributions concerning design methodology. Some of them, beyond modeling the whole design process, propose their model of the early design activities devoted to the development of the product concept, i.e. the conceptual design phase. These design approaches are widely diffused in academia. However, some uncertainties appear in literature, concerning their efficacy in performing innovative design. This observation forms the basis of this work, which aims at improving classical design processes by integrating their procedure with the TRIZ base of knowledge. To achieve such an objective, authors' approach consists in considering generally valid steps of the conceptual design process, and then in identifying most suitable TRIZ tools for each of them. A structured list of suggestions concerning the proposed integration is finally presented, together with an explanatory case study application of the proposed improvements.

Peer-review under responsibility of The International Scientific Committee ETRIA.

Keywords: TRIZ; Conceptual design; Design methods, Design models

1. Introduction

Product planning phase (PP) and early design activities are acknowledged in literature to play a critical role for the success of the product [1]. Indeed both the development stages, although in different ways, are devoted to the definition of the fundamentals of the system, strongly influencing performance, production technologies, and in general all the characteristics of the final product. The main outcome of PP is the definition of a first product requirements list that constitutes the starting point of the design process. Considering well known literature models of the design process [2, 3], the first phase is the so-called "conceptual design" (CD), which produces the definition of functionalities, working principles and a rough layout of the system structure.

The most acknowledged CD methods put their bases into functional decomposition and morphological composition of the concept, in order to define respectively the function structure of the product and the set of partial solutions which, combined together, constitute the building blocks of the system. Critics on these methods have been raised in literature, some of them concerning the real capability of such approaches in developing innovative design [4]. In order to overcome such a critical limitation, attempts devoted to upgrade classical CD processes can be found in literature, and some of them consider the TRIZ base of knowledge [5, 6] as a potential resource. Such literature contributions constitute a valid reference for this research activity; however, their general validity has not been comprehensively demonstrated.

Here arises the objective of the present work, i.e. to propose a generally valid improvement of functional decomposition and morphology based CD processes, focused on increasing their capability in developing innovative design. In order to achieve such an objective, this work points toward the identification of a set of specific TRIZ tools to be used in the main general phases of the concept development.

Section 2 proposes a short introduction to the most acknowledged CD methods with the aim of highlighting the general steps to be improved with TRIZ. Section 3 describes the research methodology and the list of identified TRIZ tools for CD. In order to validate the results, a case study application is depicted in Section 4, while discussions about the results are provided in Section 5. Eventually, concluding remarks constitute the contents of Section 6.

Peer-review under responsibility of The International Scientific Committee ETRIA.

INTERNATIONAL DESIGN CONFERENCE - DESIGN 2014
Dubrovnik - Croatia, May 19 - 22, 2014.

INVESTIGATING ON THE RISE OF MODULARITY DURING THE CONCEPTUAL DESIGN PHASE.

L. Fiorineschi, P. Rissone and F. Rotini

Keywords: modularity, conceptual design, design process

1. Introduction

According to the model proposed by [Pahl, Beitz 2007] and several other scholars, the conceptual design can be considered the phase of the design process where functionalities, physical principles, and preliminary sketches of the system physical structure are defined. The importance of the recalled design phase is well acknowledged by the literature, especially in reference to the impact it has on the overall design costs [Akay, Kulak, Henson 2011] as well as on the success probability of the design outcomes [Ulrich and Eppinger 2003]. Moreover, since the way in which functions are allocated in the components constitutes an important part of the architecture of the product [Ulrich and Eppinger 2003], it is possible to assert that defining a product concept means also defining a draft of its architecture.

Despite the presence of side effects, literature acknowledges modular type architectures to give rise to a series of positive effects. In confirmation of this, several contributions aimed at supporting the designer in reorganizing the product architecture towards modular configurations can be found in literature. However, advantages given by considering modularity early in the design process have been already inferred [Stock et al. 2003], [Graedel, Allenby 1995].

As claimed by [Hölttä-Otto 2005], [Borjesson 2010] and [Daniilidis et al. 2011], the Design Structure Matrix (DSM) [Eppinger, Browning 2012], the Function Structure Heuristics (FSH) [Stone, Wood, Crawford 2000] and the Modular Function Deployment (MFD) [Ericsson, Erixon 1999] can be considered the representative sample of the most acknowledged methods for assisting product modularization. However, it is worth of noting that the cited contributions have been developed to suggest modular reconfigurations of an existent product concept. Indeed, according to [Daniilidis et al. 2011], since the DSM-based modularization methods use a component-based analysis of the product architecture, it cannot be used until product components are determined. Concerning FSH, as stated by [Van Wie et al. 2001] the method can be used during concept design for determining the product architecture. In fact it is based on the Energy, Material and Signal (EMS) functional model [Pahl, Beitz 2007], and uses three euristics based on the EMS flows to suggest potential modules. However, the physical principles implementing the functions must be already known otherwise it results impossible to identificate the EMS flows needed for the application of the heuristics. Lastly, the MFD method has been developed to suggest potential grouping of the technical solutions composing the product, but do not give any support for their identification.

In consideration of these evidences, it is possible to claim that the results obtained by the application of the considered methods are strongly influenced by the solutions adopted in the starting concept which however has been developed without considering modularity. In fact, it can be inferred that the available modularization methods can assist concept design activities only with a trial and error approach, where a product concept has to be formerly developed. Nevertheless, [Fixson 2003] observed that the need for a modular architecture may arise during the definition of the requirement

Modularization vs. Innovation

Lorenzo Fiorineschi, Paolo Rissone, Federico Rotini
University of Florence, Industrial Engineering Department
Via Santa Marta, 3 - 50139 Firenze, Italy
lorenzo.fiorineschi@unifi.it, paolo.rissone@unifi.it, federico.rotini@unifi.it

ABSTRACT

This paper argues the relationship between modularity and product innovation. The work is based on the assumption that in order to become an innovation, a novel product has to be successfully diffused into the marketplace. Modularity can give rise to a series of parameters related to commercial success; however, there is not a well-defined relationship between modularity and product innovativeness. The aim of the paper is to analyse the logic of the most acknowledged modularization methods in-order to understand how they can really influence product success, and then, part of product innovativeness.

1. INTRODUCTION

In recent years, the term *innovation* has been widely used both in industry and in academia, even if some confusion can be observed due to the absence of a unique definition. Very often the term *invention* is associated with innovations, and for this reason the level of novelty and patentability are often used as metrics to identify innovations. However, every new idea or invention, in order to become an innovation, has to be successfully diffused into the marketplace, and economic and production tasks have to be carefully processed [1, 2].

Two different types of innovation – radical and incremental — are frequently mentioned [3]. Actually, this is a simplification, since intermediate or alternative types have also been defined [1, 4]; however, such a simplification can be considered acceptable for the scope of this paper. Both innovation types can be considered as possible outcomes of a new product development process, where a design activity has been pursued in order to obtain something based on new technological or scientific evolutions (radical innovations) or something to which new functionalities are added by means of combinations of existing solutions (incremental innovations).

When facing a novel product design process, an accurate concept definition must always be performed [5, 6]. Indeed, during conceptual design, functionalities, working principles and first sketches of the working structure of the product are developed in order to fulfill a given set of requirements. The definition of these product features strongly influences the successive design steps, and this is why conceptual design can be considered fundamental to product success.

The outcome of the concept design process can be visualized in terms of functions and a first draft of the physical structure. It is acknowledged that the way in which functions are allocated in the physical components, constitutes an important part of the product architecture, together with the definition of interfaces and the arrangement of functions [6]. During the last two decades, many contributions concerning a specific type of architecture, such as the modular one, have been written, including literature reviews dedicated to the comparison and the classification of existing definitions and methods. The interest of engineering design scholars towards modularity is motivated by the common assumption that, despite the existence of potential side effects, it can give rise to several benefits (e.g. [7]). More precisely, these benefits can lead to the reduction of costs involved in design, production, and retirement life cycle phases, and may give rise to customer satisfaction. All of these aspects are fundamental to product success and the innovativeness of the product; however, a high level of modularity can even be detrimental, due to the existence of side effects.

This conflicting behavior has been addressed in literature, highlighting that if modularity is maintained within a certain level, positive effects can be observed on product innovativeness; but beyond that level, modularity may reduce them [8].

Many attempts have been made during the last two decades in order to develop modularization methods and tools. However, a careful literature analysis has revealed that three main modularization approaches can be considered as the most representative of the current state of the art, since they are

TRIZ Future 2013

Techno-economic classification of contradictions and related strategies of solution

Gaetano Cascini[a], Lorenzo Fiorineschi[b], Francesco Saverio Frillici*[b], Paolo Rissone[b]

[a] *Politecnico di Milano, Dipartimento di Meccanica, Italy*
[b] *University of Florence, Dipartimento di Ingegneria Industriale, Italy*

Abstract

One of the most important objectives of modern product development is the fulfillment of the requirements derived from stakeholder/customer needs. For this reason, modern design processes start from an accurate definition of those final product features able to satisfy a given set of customer needs. However, it is well acknowledged that, during a common design process, it is often possible to find requirements conflicting with each other. Thus the choice of a successful design strategy is critical. The aim of this work is to investigate the possibility to find a rule suitable to indicate the best side of the contradiction to process in order to solve technical problems, also usable by engineers with limited experience with TRIZ. The analysis has been formerly operated on well-known solved problems belonging to Classical TRIZ literature; the emerging evidences have been further checked on a set of case studies from the authors' industrial experience.

Keywords: Design Requirements, Contradictions, ARIZ, Design Process

1. Introduction

The high level of competition which characterizes the modern market makes the fulfillment of customer needs as one of the most important objectives of product development. This is the reason why modern engineering problems are normally faced by addressing the fulfillment of system requirements, derived from the diverse stakeholder needs. Then, it is possible to assert that an accurate definition of their specification heavily influences the final characteristics and configurations of a product. Furthermore, in design processes it is possible to ascertain that requirements often conflict with each other, i.e. the fulfillment of a requirement often implies detrimental effects on other aspects of a product.

* Corrisponding author. Tel.: +39-055-479-6514 ; fax: +39-055-479-6525
E-mail address: francescosaverio.frillici@unifi.it

ETRIA EUROPEAN TRIZ ASSOCIATION

TRIZ Future 2012, Lisbon, Portugal

Product Architecture definition: evaluating the potentiality of TRIZ tools

Lorenzo Fiorineschi[a*], Francesco Saverio Frillici[a], Paolo Rissone[a], Gaetano Cascini[b]

[a] *University of Florence, Dipartimento di Meccanica e Tecnologie Industriali, Italy*
[b] *Politecnico di Milano, Dipartimento di Meccanica, Italy*

Abstract

Product Architecture definition plays more and more a crucial role for enhancing product customizability, easing after-sale management and reducing manufacturing costs. Despite major efforts have been dedicated to the development of methods and tools supporting Product Architecture definition for "Adaptive Design" tasks, no real means are available while addressing more radical innovation activities. The paper proposes a critical overview of TRIZ models and tools to evaluate their potential integration into a comprehensive methodology for Product Architecture definition. A comparison with the three major modularity methods is performed with the aim to establish how TRIZ can be located thereupon to current state of the art of Product Architecture management. An academic case study is discussed, in order to show how the OTSM Network of Problems approach can bring a significant contribution in that sense.

Keywords: TRIZ ; Modularity; Product Architecture; Original Design

Appendix E – Awards